



TUGAS AKHIR - TE 145561

**TELEMETERI *LEVEL* UNTUK TANGKI PENAMPUNGAN PADA
PROSES PENJERNIHAN AIR SUNGAI BERBASIS IOT
MENGUNAKAN ARDUINO**

Rivaldy Hariansyah
NRP 2214030008
Achmad Luki Satriawan
NRP 2214030112

Dosen Pembimbing
Ir. Josaphat Pramudijanto, M.Eng.

PROGRAM STUDI KOMPUTER KONTROL
Departemen Teknik Elektro Otomasi
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember
Surabaya 2017

-----Halaman ini sengaja dikosongkan-----



FINAL PROJECT - TE 145561

***LEVEL TELEMETRY FOR STORAGE TANK ON WATER
RIVER TREATMENT BASED ON IOT USING ARDUINO***

Rivaldy Hariansyah
NRP 2214030008
Achmad Luki Satriawan
NRP 2214030112

Supervisor
Ir. Josaphat Pramudijanto, M.Eng.

*COMPUTER CONTROL STUDY PROGRAM
Electrical and Automation Engineering Department
Vocational Faculty
Institut Teknologi Sepuluh Nopember
Surabaya 2017*

-----Halaman ini sengaja dikosongkan-----

PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul **“Telemetri Level Untuk Tangki Penampungan Pada Proses Penjernihan Air Sungai Berbasis IoT Menggunakan Arduino”** adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka.

Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.



Rivaldy Hariansyah
NRP 2214030008

Surabaya, 19 Juli 2017



Achmad Luki S
NRP 2214030112

-----Halaman ini sengaja dikosongkan-----

**TELEMETRI LEVEL UNTUK TANGKI PENAMPUNGAN
PADA PROSES PENJERNIHAN AIR SUNGAI BERBASIS IOT
MENGUNAKAN ARDUINO**

TUGAS AKHIR

**Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Ahli Madya
Pada
Bidang Studi Komputer Kontrol
Departemen Teknik Elektro Otomasi
Institut Teknologi Sepuluh Nopember**

Menyetujui:

Dosen Pembimbing

Ir. Josaphat Pramudijanto, M.Eng.
NIP. 1962 10 05 1990 03 1003

**SURABAYA
JULI, 2017**

-----Halaman ini sengaja dikosongkan-----

TELEMETERI *LEVEL* UNTUK TANGKI PENAMPUNGAN PADA PROSES PENJERNIHAN AIR SUNGAI BERBASIS IOT MENGUNAKAN ARDUINO

Nama : Rivaldy Hariansyah
NRP : 2214 030 008
Nama : Achmad Luki Satriawan
NRP : 2214 030 112
Pembimbing : Ir. Josaphat Pramudijanto, M.Eng.
NIP : 19621005 199003 1 003

ABSTRAK

Banyaknya unit pada industri air bersih menyebabkan diperlukan banyak operator. Hal ini kurang efisien sehingga diperlukan suatu sistem kontrol untuk menjalankan proses produksi secara otomatis. Pada proses penjernihan, diperlukan pengecekan kondisi air di laboratorium untuk mengetahui tingkat kekeruhan air sungai sehingga diperlukan sistem untuk mengecek kondisi air secara *realtime*. Pengukuran ketinggian pada tangki penampungan akhir diperlukan untuk mendeteksi banyaknya air yang diproduksi. Penggunaan *smartphone* mempermudah bagi orang-orang dalam mengakses informasi dimana saja dengan terkoneksi jaringan *internet*.

Berdasarkan hal tersebut maka dibuat alat Telemetri proses penjernihan air sungai yang didalamnya terdapat parameter yang diukur menggunakan *sensor* kekeruhan untuk membaca bahan tersuspensi pada air dan *sensor* ketinggian air untuk membaca ketinggian air pada tangki penampungan. Hasil pembacaan *sensor* dikelola Arduino lalu dikirim ke *webserver* dan di tampilkan pada Aplikasi *Android* dan LabVIEW.

Hasil pengujian Telemetri pada alat menunjukkan *sensor* kekeruhan mempunyai *error* sebesar 0,76% dan hasil pengukuran menunjukkan air bersih memiliki tingkat kekeruhan antara 5 hingga 6 NTU dan tegangan *sensor* sebesar 4,7 Volt. Hasil pembacaan *sensor* ketinggian rata-rata penurunan tegangan setiap 1 Cm 2,74 Volt dan 2,8 Volt. Pada Pengujian pengiriman data ke *server Thingspeak* memiliki *delay* antara 15 hingga 27 detik.

Kata Kunci : *Smartphone, Internet* , Penjernihan air, Telemetri

-----Halaman ini sengaja dikosongkan-----

LEVEL TELEMETRY FOR STORAGE TANK ON WATER RIVER TREATMENT BASED ON IOT USING ARDUINO

Student's Name : Rivaldy Hariansyah
Registration Number : 2214 030 008
Student's Name : Achmad Luki Satriawan
Registration Number : 2214 030 112
Supervisor : Ir. Josaphat Pramudijanto, M.Eng.
ID : 19621005 199003 1 003

ABSTRACT

The number of units in the water industry has caused many operators. It is less efficient so it needs a control system to run the production process automatically. In the process of clarification, checking the condition of the water in the laboratory to determine the level of turbidity of the river water so that the system is needed to check the water condition in realtime. Height measurements in the final shelter tanks are needed to detect the amount of water produced. The use of smartphones make it easy for people to access information anywhere with an Internet connection connected

Pursuant to that matter hence made a tool of water purifier which is as parameter turbidity sensor and water level. The sensor output is managed by arduino then the result is sent to the PC with labview as HMI and sending data using Ethernet hardware in realtime. Users can also monitor changes via Hp Android smartphone connected to internet network.

The result of telemetry testing on the tool shows that the turbidity sensor has an error of 0.76% and the measurement results show that the water has a turbidity level between 5 to 6 NTU and a sensor voltage of 4.7 Volt. Results of the average height of the sensor readings decrease the voltage every 1 Cm 2.74 Volt and 2.8 Volt. On Testing data delivery to the server Thingspeak has a delay between 15 to 27 seconds.

Keywords : Smartphone, Internet, Purifying water

-----Halaman ini sengaja dikosongkan-----

KATA PENGANTAR

Puji syukur penulis panjatkan kehadiran Allah SWT yang selalu memberikan rahmat dan hidayah-Nya sehingga Tugas Akhir ini dapat terselesaikan dengan baik. Shalawat serta salam semoga selalu dilimpahkan kepada Rasulullah Muhammad SAW, keluarga, sahabat, dan umat muslim yang senantiasa meneladani beliau.

Tugas Akhir ini disusun untuk memenuhi sebagian persyaratan guna menyelesaikan pendidikan Diploma III pada Bidang Studi Komputer Kontrol, Departemen Teknik Elektro Otomasi, Fakultas Teknik, Institut Teknologi Sepuluh Nopember Surabaya dengan judul:

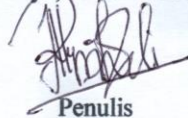
TELEMETRI LEVEL UNTUK TANGKI PENAMPUNGAN PADA PROSES PENJERNIHAN AIR SUNGAI BERBASIS IOT MENGGUNAKAN ARDUINO.

Dalam Tugas Akhir ini dirancang untuk membuat *prototype* alat penjernihan air yang ditelemetering ke *server* berbasis IoT dengan *Arduino* sebagai kontroler.

Penulis mengucapkan terima kasih kepada Ibu dan Bapak penulis yang memberikan berbagai bentuk doa serta dukungan tulus tiada henti, Ir. Josaphat Pramudijanto, M.Eng. atas segala bimbingan ilmu, moral dan spiritual dari awal hingga terselesaikannya Tugas Akhir ini. Penulis juga mengucapkan banyak terima kasih kepada semua pihak yang telah membantu baik secara langsung maupun tidak langsung dalam proses penyelesaian Tugas Akhir ini.

Penulis menyadari dan memohon maaf atas segala kekurangan pada Tugas Akhir ini. Akhir kata, semoga Tugas Akhir ini dapat bermanfaat dalam pengembangan keilmuan di kemudian hari.

Surabaya, 19 Juli 2017



Penulis

-----Halaman ini sengaja dikosongkan-----

DAFTAR ISI

HALAMAN

HALAMAN JUDUL.....	i
HALAMAN JUDUL.....	iii
PERNYATAAN KEASLIAN TUGAS AKHIR	v
LEMBAR PENGESAHAN.....	vii
ABSTRAK	ix
<i>ABSTRACT</i>	xi
KATA PENGANTAR.....	xiii
DAFTAR ISI.....	xv
DAFTAR GAMBAR	xix
DAFTAR TABEL.....	xxiii
 BAB I PENDAHULUAN	 1
1.1 Latar Belakang.....	1
1.2 Permasalahan	3
1.3 Batasan Masalah	3
1.4 Tujuan.....	4
1.5 Sistematika Laporan	4
1.6 Relevansi	6
 BAB II TEORI DASAR.....	 7
2.1 Air	7
2.1.1 Persyaratan Air Bersih Secara Fisik	7
2.1.2 Persyaratan Air Bersih Secara Kimiawi	8
2.1.3 Parameter Air Bersih Secara Radiologi.....	8
2.1.4 Proses Penjernihan Air Sungai	8
2.2 <i>Sensor</i> Kekeruhan	10
2.3 SCADA	11
2.4 <i>LabVIEW Software</i>	12
2.4.1 <i>Front Panel</i>	13
2.4.2 Diagram Blok	14
2.5 <i>Webserver</i>	14
2.6 Arduino Uno	15
2.6.1 Spesifikasi Arduino	16
2.7 <i>Relay</i>	18
2.8 Telemetry	19
2.9 <i>Internet of Things</i>	20
2.10 <i>Thingspeak</i>	22
2.11 <i>Ethernet Shield</i>	22

2.12	<i>Buck Converter</i>	23
2.13	<i>App Inventor</i>	24
2.14	Motor DC	25
2.15	<i>Solenoid Valve</i>	25
2.16	<i>Sensor Ketinggian Air</i>	27
BAB III PERANCANGAN SISTEM		29
3.1	Diagram Fungsional Alat.....	29
3.2	Perencanaan dan Pembuatan <i>Hardware</i>	32
3.2.1	Perancangan Sistem <i>Input/ Output</i> Modul Arduino	32
3.2.2	Perancangan <i>Sensor Ketinggian</i>	33
3.2.3	Perancangan <i>Sensor Kekeruhan</i>	35
3.2.4	Perancangan Besi Penyangga	36
3.2.5	Perancangan <i>Hardware</i> Aquarium	37
3.2.6	Perancangan Tangki Pengadukan Dan Tangki Final.....	38
3.2.7	Perancangan <i>Sensor Ketinggian Air</i>	38
3.3	Perancangan <i>Software</i>	39
3.3.1	Perancangan Sistem Secara Keseluruhan	39
3.3.2	Perancangan <i>Software Sensor Kekeruhan</i>	43
3.3.3	Perancangan <i>Software Sensor Ketinggian</i>	44
3.3.4	Perancangan <i>Software Ethernet Shield</i>	45
3.3.5	Perancangan <i>Software HMI LabVIEW</i>	45
3.3.6	Perancangan <i>Software Thingspeak</i>	53
3.3.7	Perancangan <i>Website</i>	55
3.3.8	Perancangan Aplikasi <i>Android</i>	62
3.3.9	Perancangan Sistem Telemetry <i>Level</i> Dan Kekeruhan Air Pada Alat Penjernihan Air Sungai.....	63
BAB IV HASIL SIMULASI DAN UJI COBA		65
4.1	Pengujian Pompa Air	65
4.2	Pengujian <i>Sensor Ketinggian</i>	66
4.2.1	Perhitungan <i>Volume</i> Tangki	74
4.2.2	Rangkaian Jembatan <i>Wheatstone</i>	75
4.2.3	Rangkaian <i>Signal Conditioning</i>	77
4.3	Pengujian <i>Relay</i>	79
4.4	Pengujian Motor DC	81
4.5	Pengujian <i>Sensor Kekeruhan</i>	82
4.6	Pengujian <i>Solenoid Valve</i>	87
4.7	Pengujian HMI LabVIEW	89
4.8	Pengujian pada <i>Webserver</i>	93
4.9	Pengujian Aplikasi <i>Android</i>	105

4.10 Pengujian Proses Penjernihan Air.....	108
4.11 Pengujian Alat Secara Keseluruhan.....	110
BAB V PENUTUP.....	117
5.1 Kesimpulan.....	117
5.2 Saran.....	117
DAFTAR PUSTAKA	119
LAMPIRAN A	121
A.1 Lampiran Program Keseluruhan	121
A.2 Lampiran Program Pengujian Pompa Air dan <i>Relay</i>	131
LAMPIRAN B	133
B.1 Kontroler Alat.....	133
B.2 Alat Telemetry Penjernihan	133
B.3 Alur Kerja dari Alat Sebelum Filter.....	134
B.4 Alur Kerja dari Alat Setelah Filter.....	134
B.5 Kotak Kontroler.....	136
B.6 Filter Aquarium	136
B.7 <i>Sensor</i> Kekeruhan Pada Alat	137
B.8 Proses Pengisian	137
LAMPIRAN C	139

-----Halaman ini sengaja dikosongkan-----

DAFTAR GAMBAR

HALAMAN

Gambar 2.1	<i>Sensor Kekeruhan</i>	11
Gambar 2.2	<i>Front Panel</i> Dan Diagram Blok	14
Gambar 2.3	Arduino Uno	16
Gambar 2.4	<i>Relay 4 Channel</i>	19
Gambar 2.5	Diagram Blok Telemetry	20
Gambar 2.6	<i>IoT Clouds</i>	21
Gambar 2.7	Halaman Awal <i>Website Thingspeak</i>	22
Gambar 2.8	<i>Ethernet Shield</i>	23
Gambar 2.9	Rangkaian <i>Buck Converter</i>	24
Gambar 2.10	<i>Buck Converter</i>	24
Gambar 2.11	Aplikasi <i>App Inventor Android</i>	25
Gambar 2.12	Motor DC	26
Gambar 2.13	<i>Solenoid Valve</i>	26
Gambar 2.14	<i>Sensor Ketinggian Air</i>	27
Gambar 3.1	Diagram Fungsional <i>Prototype</i> Tugas Akhir	30
Gambar 3.2	Perancangan <i>Hardware</i> Tampak Dalam	32
Gambar 3.3	Perancangan <i>Hardware</i> Tampak Luar	32
Gambar 3.4	Perancangan I/O Arduino	34
Gambar 3.5	Perancangan <i>Sensor Ketinggian</i>	35
Gambar 3.6	Perancangan <i>Sensor Kekeruhan</i>	36
Gambar 3.7	Bentuk Fisik Wadah <i>Sensor Kekeruhan</i>	36
Gambar 3.8	Perancangan Penyangga Aquarium	37
Gambar 3.9	Perancangan Penyangga Panel Kontrol	37
Gambar 3.10	Perancangan Mekanik Aquarium Untuk Proses Filtrasi	38
Gambar 3.11	Perancangan Tangki	38
Gambar 3.12	Perancangan Mekanik <i>Sensor Ketinggian Air</i>	39
Gambar 3.13	<i>Flowchart</i> Alat Secara Umum	41
Gambar 3.14	<i>Flowchart</i> Proses Penjernihan Air	42
Gambar 3.15	<i>Flowchart Sensor Kekeruhan</i>	43
Gambar 3.16	<i>Flowchart Sensor Ketinggian</i>	44
Gambar 3.17	<i>Flowchart</i> Kerja <i>Ethernet Shield</i>	46
Gambar 3.18	<i>Block Diagram</i> Mengakses Data <i>Sensor</i>	47
Gambar 3.19	<i>Block Diagram</i> Mengakses Data Aktuator	48
Gambar 3.20	<i>Block Diagram</i> Mengirim Data Aktuator	49
Gambar 3.21	HMI LabVIEW Pengaturan Alat	50
Gambar 3.22	HMI LabVIEW Grafik <i>Sensor</i>	51

Gambar 3.23	<i>Flowchart LabVIEW</i>	52
Gambar 3.24	Pembuatan <i>Channel</i>	53
Gambar 3.25	Mengisi Data <i>Channel</i>	54
Gambar 3.26	Hasil Pembacaan <i>Sensor</i> Oleh <i>Thingspeak</i>	57
Gambar 3.27	<i>Script</i> Pada Notepad+	58
Gambar 3.28	Aplikasi FileZilla Untuk Mengirim Data ke <i>Server</i>	59
Gambar 3.29	<i>Phpmyadmin</i> Untuk Membuat <i>Database MySQL</i>	60
Gambar 3.30	Halaman <i>Website</i> Setelah Diupload	61
Gambar 3.31	Antar Muka <i>Android</i>	62
Gambar 3.32	<i>Block Editor Android</i>	63
Gambar 3.33	Skema Alur Pembuatan <i>Android</i>	63
Gambar 3.34	Rancangan Alat Keseluruhan	64
Gambar 4.1	Rangkaian Pengujian Pompa	66
Gambar 4.2	Rangkaian Pengujian <i>Sensor</i> Ketinggian	67
Gambar 4.3	Bentuk <i>Hardware Sensor</i> Ketinggian	67
Gambar 4.4	Pengujian <i>Sensor</i> Ketinggian Pada Tangki Penampungan Awal	70
Gambar 4.5	Pengujian <i>Sensor</i> Ketinggian Pada Tangki Penampungan Akhir.....	73
Gambar 4.6	Limas Terpancung	74
Gambar 4.7	Rangkaian Jembatan <i>Wheatstone</i>	76
Gambar 4.8	Rangkaian <i>Signal Conditioning</i>	77
Gambar 4.9	Pengujian <i>Relay</i>	80
Gambar 4.10	Kontruksi Motor DC pada Tangki Air	81
Gambar 4.11	Pengujian Motor DC	82
Gambar 4.12	Rangkaian Pengujian <i>Sensor</i> Kekeruhan	83
Gambar 4.13	Pengujian Kekeruhan Air Menggunakan <i>Sensor</i> Kekeruhan	84
Gambar 4.14	<i>Hardware Sensor</i> Kekeruhan Pada <i>Prototype</i>	84
Gambar 4.15	Air Untuk Pengujian Tingkat Kekeruhan	85
Gambar 4.16	Alat Ukur <i>Turbidity Meter</i> merk Oakton T100.....	85
Gambar 4.17	Pengujian Kekeruhan Air Menggunakan <i>Turbidity</i> <i>Meter</i>	86
Gambar 4.18	Pengujian Alat Ukur Kekeruhan Air.....	87
Gambar 4.19	Pengujian <i>Solenoid Valve</i>	88
Gambar 4.20	Letak <i>Solenoid Valve</i> Pada Alat.....	88
Gambar 4.21	HMI LabVIEW Ketika Di Jalankan.....	90
Gambar 4.22	HMI LabVIEW Setelah Mengambil Data Dari <i>Server</i> ...	91
Gambar 4.23	HMI LabVIEW Setelah Mengirim Data ke <i>Server</i>	92
Gambar 4.24	<i>Login Website</i>	96

Gambar 4.25	Pengukuran <i>Sensor</i> Kekeruhan Pada <i>Website</i>	97
Gambar 4.26	Pengukuran <i>Sensor</i> Ketinggian Pada <i>Website</i>	98
Gambar 4.27	Data Kondisi Aktuator Pada <i>Website</i>	99
Gambar 4.28	Data Pengukuran <i>Sensor</i> Pada <i>Website</i>	100
Gambar 4.29	Kondisi Aktuator.....	101
Gambar 4.30	Mengubah Kondisi Mode Dari Aktuator	102
Gambar 4.31	Ketika Data Telah Berhasil Dikirim Ke <i>Server</i>	103
Gambar 4.32	Mengubah Konfigurasi Aktuator	104
Gambar 4.33	Data <i>Sensor</i> Di <i>Android</i>	107
Gambar 4.34	<i>Report</i> Data Pengukuran <i>Sensor</i> Di <i>Android</i>	107
Gambar 4.35	Kondisi Air Setelah Diendapkan Selama 2 Hari	109
Gambar 4.36	Kondisi Air Setelah Dicampurkan dengan Tawas	109
Gambar 4.37	Alat Telemetry Proses Penjernihan Air	112

-----Halaman ini sengaja dikosongkan-----

DAFTAR TABEL

HALAMAN

Tabel 2.1	Spesifikasi Arduino Uno.....	17
Tabel 3.1	Konfigurasi Pengisian <i>Channel</i>	54
Tabel 4.1	Data Pengujian Pompa Air.....	66
Tabel 4.2	Data Pengujian Tangki Penampungan Awal.....	68
Tabel 4.3	Data Ketinggian Tangki Penampungan Awal.....	69
Tabel 4.4	Data Hasil Pengujian Tangki Akhir	71
Tabel 4.5	Data Perbandingan Ketinggian Tangki Penampungan Akhir.....	72
Tabel 4.6	Pengukuran Tegangan <i>output J.Wheatstone</i> Tangki Penampungan Awal	75
Tabel 4.7	Data Pengukuran Tegangan <i>output J Wheatstone</i> Tangki Akhir.....	76
Tabel 4.8	Data Pengukuran Rangkaian <i>Signal Conditioning</i> Tangki Awal	78
Tabel 4.9	Data Pengukuran Rangkaian <i>Signal Conditioning</i> Tangki Akhir.....	79
Tabel 4.10	Data Hasil Pengujian <i>Relay</i>	80
Tabel 4.11	Data Hasil Pengujian Motor DC Tanpa Beban	82
Tabel 4.12	Data Hasil Pengujian Motor DC Dengan Beban.....	82
Tabel 4.13	Data Hasil Pengujian <i>Sensor</i> Kekakuan	86
Tabel 4.14	Data Hasil Pengujian <i>Solenoid Valve</i>	88
Tabel 4.15	Alamat Halaman <i>Website</i> Berdasarkan <i>Icon</i>	93
Tabel 4.16	Data Excel <i>Monitoring Actuator</i>	94
Tabel 4.17	Data Penomoran <i>Field</i>	95
Tabel 4.18	Data Pengiriman Ke <i>Server Thingspeak</i>	105
Tabel 4.19	Data Pengujian Aplikasi <i>Android</i>	106
Tabel 4.20	Hasil Pengujian Penjernihan Air.....	108
Tabel 4.21	Hasil Pengujian Pembacaan <i>Sensor</i> Pada Alat.....	112
Tabel 4.22	Hasil Pengujian Kondisi Aktuator Pada Alat.....	113

-----Halaman ini sengaja dikosongkan-----

BAB I

PENDAHULUAN

1.1 Latar Belakang

Didalam proses industri penyedia air bersih terdapat beberapa tahapan yang dilakukan yang masing-masing tahapan tersebut terdapat pada *plant* antara lain bangunan *intake*, tangki pencampuran bahan kimia, filterisasi dan tangki *reservoir*. Pada *plant intake* berguna sebagai menyaring kotoran besar seperti sampah agar tidak tercampur pada proses dan *plant* tangki pencampuran memiliki fungsi pencampuran bahan kimia seperti tawas. Pada tahap filterisasi berguna dalam menyaring gumpalan-gumpalan yang terbentuk akibat proses pencampuran tawas dan akan melakukan proses pengendapan. Banyaknya unit pada industri air bersih menyebabkan diperlukan banyak operator untuk mengoperasikan. Hal ini kurang efisien sehingga diperlukan suatu sistem kontrol untuk menjalankan proses produksi secara otomatis. Sistem otomatis ini dapat mengurangi jumlah operator dalam mengoperasikan alat namun terbatas pada hal *monitoring* saja. Penggunaan sistem kontrol dapat menggantikan peran pekerja dalam menjalankan proses pengontrolan pada setiap *plant* serta dapat mempercepat pengambilan keputusan dalam pengoperasian alat.

Kualitas air bersih yang dihasilkan bergantung pada proses penjernihan air yang dilakukan oleh perusahaan penyedia air bersih. Perusahaan penyedia air bersih akan melakukan beberapa tahap untuk proses penjernihan air. Air dikatakan bersih dengan kriteria tidak keruh, tidak berwarna, serta tidak memiliki rasa dan bau. Kegunaan air pada proses ini berguna untuk kebutuhan sehari-hari seperti untuk cuci-cuci, mandi dan kebutuhan sehari-hari yang lain. Untuk menjaga produksi air bersih memiliki kualitas yang baik, pada setiap *plant* terdapat petugas untuk mengawasi seperti mengontrol dan memantau kondisi air, salah satu *plant* yang perlu dilakukan pengawasan intensif adalah tangki pencampuran dan tangki penampungan produk air bersih. Seringkali *volume* air yang ditampung pada tempat penampungan akhir melebihi *volume* maksimal air yang ditampung sehingga seringkali air meluber dan *sensor* air yang digunakan pada tempat penampungan air tidak berfungsi dengan baik dan seringkali terdapat keluhan dari masyarakat mengenai air yang dialirkan ke rumah-rumah tidak berjalan dengan lancar, masyarakat mengadu ke perusahaan penyedia air untuk

dilakukan penanganan yang benar. Dengan pengaduan seperti ini, perusahaan penyedia air memerlukan suatu metode yang dapat memantau debit air yang di alirkan ke rumah masyarakat apakah air mengalami kebocoran pipa atau penggunaan air yang berlebihan dari masyarakat sehingga pihak penyedia perusahaan memerlukan metode Telemetri untuk memantau kondisi ketinggian air secara *realtime* dari *local control room*.

Sebelum melalui proses produksi, dilakukan pengujian pada kualitas air baku secara intensif oleh petugas khusus untuk menentukan penanganan yang perlu dilakukan agar memproduksi air bersih dengan kualitas yang baik. Dalam menguji kualitas air yang dilakukan, petugas akan mengambil *sample* data seperti suhu, pH, alkalinitas, kekeruhan, warna, dan oksigen terlarut. Pada pengukuran suhu dan pH berdampak pada takaran bahan kimia yang dilarutkan pada air dan berpengaruh terhadap proses koagulasi atau pembentukan gumpalan yang dilakukan. Semua parameter kimia terlarut akan dilakukan pengujian dilaboratorium untuk dianalisa kualitas air yang tidak kasat mata. Dalam menganalisa kualitas air membutuhkan waktu yang cukup lama karena dilakukan secara manual oleh petugas dan biasanya membutuhkan waktu beberapa jam dalam menguji dan menganalisa kualitas air. Hal ini menyebabkan tingkat akurasi pengukuran tidak valid. Untuk menghasilkan data pengukuran kualitas yang tepat diperlukan *sensor* yang mampu membaca keadaan lapangan. Tetapi lokasi *plant* terletak jauh dari tempat *local control room* sehingga diperlukan metode untuk pembacaan kondisi lapangan yang letaknya jauh dari operator. Untuk mengirimkan data *sensor* ke operator yang letaknya jauh dari *plant* diperlukan sistem Telemetri. Sistem Telemetri mengurangi tugas petugas khusus dalam melakukan pengujian kualitas air secara berkala. Sistem Telemetri mempercepat kinerja operator dan sistem kontrol dalam mengambil tindakan yang tepat untuk mengolah air sungai menjadi air bersih dengan kualitas air seperti menentukan kadar alum yang diperlukan atau cairan asam untuk menetralkan pH dari air sungai.

Diperlukan HMI (*Human Machine Interface*) untuk dapat memvisualkan hasil pengukuran *sensor* dan kontrol *plant* . Akan tetapi HMI hanya dapat digunakan di *control room*. Sehingga tidak dapat dilakukan *monitoring* di tempat lain maka dari itu diperlukan suatu sistem yang berbasis IoT. Sistem IoT memudahkan dalam mengakses data dari mana saja. Oleh sebab itu dibuatlah HMI untuk ponsel berbasis *Android* yang terhubung *internet*, yang mana jaringan *internet* tersebut memerlukan teknologi IoT (*Internet of Things*). IoT merupakan sebuah

konsep penggunaan konektivitas *internet* untuk menghubungkan beberapa alat / benda secara terus menerus dengan tujuan untuk berbagi data, *remote control* ataupun keperluan lain. Penerapan Sistem IoT pada industri air bersih memiliki banyak keuntungan. Kontroler dan *device* yang terdapat pada setiap *local control* dapat terhubung dan saling bertukar data secara cepat. Data dari lapangan dapat diakses darimana saja karena Sistem IoT berupa sebuah *database* yang mampu menyimpan dan mendistribusikan segala informasi yang dibutuhkan sehingga memudahkan operator ataupun atasan memantau proses dan kondisi lapangan. Petugas dapat mendapatkan informasi mengenai gangguan yang terjadi di lapangan seperti pompa berhenti bekerja ataupun *valve* tersumbat dapat langsung dilakukan analisa mengenai kondisi lapangan dan melakukan tindakan untuk menyelesaikan gangguan tersebut.

1.2 Permasalahan

Permasalahan kami menuliskan Tugas Akhir ini adalah:

- a. Banyaknya *plant* pada industri penyedia air memerlukan sistem kontrol untuk menggantikan jumlah operator yang terlalu banyak sehingga sistem dapat berjalan secara otomatis.
- b. *Monitoring* pengukuran kekeruhan air dilakukan secara manual dan dalam mengambil *sample* air membutuhkan petugas khusus untuk melakukan pengukuran.
- c. Pemantauan tingkat kekeruhan dan proses penjernihan air tidak dapat dilakukan secara jarak jauh.
- d. Pemantauan proses hanya dapat dilakukan dalam *control room* melalui HMI di Operator.

1.3 Batasan Masalah

Kami membuat batasan masalah yang akan dibahas dalam pembuatan Tugas Akhir ini, yaitu:

- a. Metode pengambilan data mengacu pada 2 buah *sensor* yaitu *sensor* ketinggian air menggunakan Potensiometer *Ten Turn* dan *sensor* kekeruhan air.
- b. Sistem hanya dirancang untuk *monitoring* ketinggian air pada tangki penampungan awal dan tangki penampungan akhir serta kekeruhan air pada sebelum dan sesudah proses penjernihan air.
- c. Pengambilan parameter *sample* air sungai jagir dilakukan di pinggir jalan yang terletak di Jl Jagir Wonokromo.

- d. Media pengiriman menggunakan kabel RJ-45 yang tersambung dengan *router* sehingga lokasi *plant* tidak dapat diubah/ditempat yang tetap.
- e. *Sensor* ketinggian yang digunakan hanya cocok pada ketinggian 25 Cm apabila ketinggian tangki diubah maka perlu dilakukan penyesuaian pada roda gigi *sensor* ketinggian.
- f. HMI yang digunakan adalah *website* melalui *browser* ,LabVIEW melalui Windows dan aplikasi penjernihan air melalui *smartphone Android*.
- g. Kontrol yang digunakan adalah kontroler ON/OFF dan hanya dapat dilakukan melalui HMI.

1.4 Tujuan

Tujuan kami menuliskan Tugas Akhir ini adalah:

- a. Membuat alat untuk proses penjernih air sungai menjadi air bersih. (Penanggung jawab Rivaldy Hariansyah)
- b. *Monitoring* ketinggian air pada proses penjernihan air menggunakan *sensor level*. (Penanggung jawab Achmad Luki Satriawan)
- c. Membuat *website monitoring* proses penjernihan air yang mampu diakses oleh *client*. (Penanggung jawab Rivaldy Hariansyah)
- d. Membuat aplikasi *Android* untuk *monitoring* proses penjernihan air. (Penanggung jawab Achmad Luki Satriawan)
- e. Membuat HMI LabVIEW sebagai media komunikasi antara arduino dengan pengguna. (Achmad Luki Satriawan bertanggung jawab membuat HMI *monitoring* ketinggian air dan Rivaldy Hariansyah bertanggung jawab membuat HMI *monitoring* tingkat kekeruhan air)

1.5 Sistematika Laporan

Pembahasan Tugas Akhir ini akan dibagi menjadi lima Bab dengan sistematika sebagai berikut:

Bab I PENDAHULUAN

Membahas tentang latar belakang, perumusan masalah, batasan masalah, maksud dan tujuan, sistematika laporan, serta relevansi Tugas Akhir yang dibuat.

Bab II TEORI PENUNJANG

Menjelaskan teori yang berisi teori-teori penunjang yang dijadikan landasan prinsip dasar dan mendukung dalam perencanaan dan pembuatan alat yang dibuat. Rivaldy Hariansyah bertanggung jawab terhadap teori penunjang yang berkaitan dengan proses penjernihan dan Telemetry tingkat kekeruhan air. Achmad Luki Satriawan bertanggung jawab terhadap teori penunjang yang berkaitan dengan Telemetry *volume* air pada tangki dan teori *Internet of Things*.

Bab III PERANCANGAN ALAT

Membahas perencanaan dan pembuatan tentang perencanaan dan pembuatan perangkat keras (*Hardware*) yang meliputi desain mekanik dan perangkat lunak (*Software*) yang meliputi *program* yang akan digunakan untuk menjalankan alat tersebut. Pada perancangan mekanik Rivaldy Hariansyah bertanggung jawab dalam mengerjakan perancangan *Sensor* kekeruhan dan pada perancangan *software* bertanggung jawab dalam perancangan keseluruhan sistem, *sensor* kekeruhan, *software website* dan HMI LabVIEW. Pada perancangan mekanik Achmad Luki Satriawan bertanggung jawab dalam mengerjakan perancangan *input/output* Modul Arduino dan *Sensor Level* serta pada perancangan *software* bertanggung jawab pada perancangan *sensor* ketinggian, *software thingspeak* dan aplikasi *Android*.

Bab IV PENGUKURAN DAN ANALISA

Membahas pengujian alat dan menganalisa data yang didapat dari pengujian tersebut serta membahas tentang pengukuran, pengujian, dan penganalisaan terhadap alat. Rivaldy Hariansyah bertanggung jawab terhadap pelaksanaan dan hasil penelitian proses penjernihan. Achmad Luki Satriawan bertanggung jawab terhadap pelaksanaan dan hasil penelitian *volume* air pada tangki.

Bab V PENUTUP

Berisi penutup yang menjelaskan tentang kesimpulan yang didapat dari Tugas Akhir ini dan saran-saran yang dapat diimplementasikan untuk pengembangan alat ini lebih lanjut.

1.6 Relevansi

Relevansi kami dalam menuliskan Tugas Akhir ini adalah:

- a. Memudahkan dalam *monitoring* parameter kualitas air.
- b. Mencegah terjadinya pencemaran pada air dengan perawatan *preventif* pada alat sehingga mutu air tetap terjaga.

BAB II

TEORI DASAR

Pada bab ini akan dibahas mengenai dasar-dasar teori yang berhubungan dengan alat seperti teori penunjang yang terdapat 16 poin. Tinjauan pustaka terbagi menjadi 2 bagian pada masing-masing mahasiswa, Rivaldy Hariansyah mencari bahan materi pada poin 2.1 sampai dengan poin 2.7 dan Achmad Luki Satriawan mencari bahan dan materi pada poin 2.8 sampai dengan poin 2.16 pada tiap materi yang dicari digunakan sebagai dasar materi untuk pembuatan alat berupa *prototype* proses penjernihan air serta sebagai dasar materi yang dibuat masing-masing mahasiswa untuk pembuatan keseluruhan alat ini.

2.1 Air [1]

Air adalah senyawa kimia yang merupakan hasil ikatan dari unsur hidrogen (H_2) yang bersenyawa dengan unsur oksigen (O) dalam hal ini membentuk senyawa H_2O . Air merupakan senyawa kimia yang sangat penting bagi kehidupan makhluk hidup di bumi ini. Fungsi air bagi kehidupan tidak dapat digantikan oleh senyawa lain. Penggunaan air yang utama dan sangat penting bagi kehidupan adalah sebagai air minum. Hal ini terutama untuk mencukupi kebutuhan air di dalam tubuh manusia itu sendiri.

Air bersih adalah salah satu jenis sumber daya berbasis air yang bermutu baik dan biasa dimanfaatkan oleh manusia untuk dikonsumsi atau dalam melakukan aktivitas mereka sehari-hari. berdasarkan standar peraturan Menteri Kesehatan RI No. 416/MENKES/PER/IX/1990 tentang Persyaratan Kualitas Air Bersih dan sehat.

2.1.1 Persyaratan Air Bersih Secara Fisik

Kualitas fisik yang dipertahankan atau dicapai bukan hanya semata-mata dengan pertimbangan dari segi kesehatan saja akan tetapi juga menyangkut keamanan dan dapat diterima oleh masyarakat pengguna air dan juga pula menyangkut segi estetika. Secara fisik air yang bersih dan sehat dengan ciri-ciri:

1. Air harus bersih dan tidak keruh.
2. Tidak berwarna apapun.
3. Tidak berasa apapun.
4. Tidak berbau apapun.
5. Suhu antara $10^{\circ} - 25^{\circ}C$
6. Tidak meninggalkan endapan.

2.1.2 Persyaratan Air Bersih Secara Kimiawi

Kandungan unsur kimia didalam air harus mempunyai kadar dan tingkat konsentrasi tertentu yang tidak membahayakan kesehatan manusia atau mahluk hidup lainnya, pertumbuhan tanaman, atau tidak membahayakan kesehatan pada penggunaannya dalam industri serta tidak menimbulkan kerusakan-kerusakan pada instalasi sistem penyediaan air minumannya sendiri. Persyaratan kimiawi antara lain yaitu:

- 1 Tidak mengandung bahan kimiawi yang mengandung racun.
- 2 Tidak mengandung zat-zat kimiawi yang berlebihan.
- 3 Cukup Yodium.
- 4 pH air antara 6,5 – 9,2.
- 5 Persyaratan Bakteriologi.

Dalam persyaratan ini ditentukan batasan tentang jumlah bakteri pada umumnya dan khususnya bakteri penyebab penyakit (ekoli).

2.1.3 Parameter Air Bersih Secara Radiologi

Parameter air bersih dan sehat secara radiologi meliputi:

- 1 Konduktifitas atau daya hantar.
- 2 Pesistifitas.
- 3 PTT atau TDS (kemampuan air bersih untuk menghantarkan arus listrik)

2.1.4 Proses Penjernihan Air Sungai

Sistem pengelolaan air dikenal dengan istilah *Water Treatment*. Ada beberapa tahap pengelolaan air yang harus dilakukan sehingga air tersebut bisa dikatakan layak untuk dipakai. Namun, tidak semua tahap ini diterapkan oleh masing-masing pengelola air, tergantung dari kualitas sumber airnya. Karakteristik umum air sungai adalah terdapat kandungan partikel tersuspensi atau koloid.

Sebagai contoh, jika sumber airnya berasal dari dalam tanah (*ground water*), sistem pengelolaan airnya akan lebih sederhana dari pada yang sumber airnya berasal dari sumber air permukaan, seperti air sungai, danau atau laut. Karena air yang berasal dari dalam tanah telah melalui penyaringan secara alami oleh struktur tanah itu sendiri dan tidak terkontak langsung dengan udara bebas yang mengandung banyak zat-zat pencemaran air. Berbeda halnya dengan sumber air permukaan yang mudah sekali tercemar. Namun demikian air yang berasal dari dalam tanah pun akan jadi tercemar juga jika sistem penampungan dan penyalurannya tidak bagus. Bila air sungai mempunyai kekeruhan atau kadar lumpur yang tinggi, maka diperlukan tambahan unit *pre-treatment*

meliputi *screen* dan *pra-sedimentasi*. Bila kadar oksigen sangat rendah maka diperlukan tambahan *unit* aerasi. Bila terdapat kandungan kesadahan yang tinggi maka diperlukan tambahan *unit* penurunan kesadahan (presipitasi dengan kapur/soda sedimentasi - rekarbonasi).

Secara umum proses pengolahan air dibagi dalam 2 *unit*, yaitu:

1. *Unit* Penampungan Awal (*Intake*)

Unit ini dikenal dengan istilah *unit* Sadap Air (*Intake*). *Unit* ini berfungsi sebagai tempat penampungan air dari sumber airnya. Selain itu *unit* ini dilengkapi dengan *Bar Screen* yang berfungsi sebagai penyaring awal dari benda-benda yang ikut tergenang dalam air seperti sampah daun, kayu dan benda lainnya.

2. *Unit* Pengolahan (*Water Treatment*)

Pada *unit* ini, air dari *unit* penampungan awal diproses melalui beberapa tahapan:

- Tahap Koagulasi (*Coagulation*)

Pada tahap ini, air yang berasal dari penampungan awal diproses dengan menambahkan zat kimia Tawas (alum) atau zat sejenis seperti zat garam besi (*Salts Iron*) atau dengan menggunakan sistem pengadukan cepat (*Rapid Mixing*). Air yang kotor atau keruh umumnya karena mengandung berbagai partikel koloid yang tidak terpengaruh gaya gravitasi sehingga tidak bisa mengendap dengan sendirinya. Tujuan dari tahap ini adalah untuk menghancurkan partikel koloid (yang menyebabkan air keruh) tadi sehingga terbentuk partikel-partikel kecil namun masih sulit untuk mengendap dengan sendirinya.

- Tahap Flokulasi (*Flocculation*)

Proses Flokulasi adalah proses penyisihan kekeruhan air dengan cara penggumpalan partikel untuk dijadikan partikel yang lebih besar (partikel Flok). Pada tahap ini, partikel-partikel kecil yang terkandung dalam air digumpalkan menjadi partikel-partikel yang berukuran lebih besar (Flok) sehingga dapat mengendap dengan sendirinya (karena gravitasi) pada proses berikutnya. Pada proses Flokulasi ini dilakukan dengan cara pengadukan lambat (*Slow Mixing*).

- Tahap Pengendapan (*Sedimentation*)

Pada tahap ini partikel-partikel flok tersebut mengendap secara alami di dasar penampungan karena massa jenisnya

lebih besar dari unsur air. Kemudian air di alirkan masuk ke tahap penyaringan di *Unit Filtrasi*.

- Tahap Penyaringan (*Filtration*)
Pada tahap ini air disaring melewati media penyaring yang disusun dari bahan-bahan berupa pasir dan kerikil Silica. Proses ini ditujukan untuk menghilangkan bahan-bahan terlarut dan tak terlarut. Secara umum setelah melalui proses penyaringan ini air langsung masuk ke *unit* Penampungan Akhir. Namun untuk meningkatkan kualitas air kadang diperlukan proses tambahan, seperti: Proses Pertukaran *ion* (*Ion Exchange*), Proses Penyerapan (*Absorption*), Proses Disinfeksi (*Disinfection*), dan *Unit* Penampung Akhir (*Reservoir*). Proses pertukaran *ion* bertujuan untuk menghilangkan zat pencemar anorganik yang tidak dapat dihilangkan oleh proses filtrasi atau sedimentasi. Proses Penyerapan ini bertujuan untuk menyerap / menghilangkan zat pencemar organik, senyawa penyebab rasa, bau dan warna. Biasanya dengan membubuhkan bubuk karbon aktif ke dalam air tersebut. Sebelum masuk ke *unit* Penampungan Akhir, air melalui Proses Disinfeksi dahulu. Yaitu proses pembubuhan bahan kimia *Chlorine* yang bertujuan untuk membunuh bakteri atau mikroorganisme berbahaya yang terkandung di dalam air tersebut setelah masuk ke tahap *Unit* Penampung Akhir berarti air sudah siap untuk didistribusikan ke masyarakat.

2.2 *Sensor* Kekeruhan [2]

Sensor kekeruhan adalah salah satu alat umum yang biasa digunakan untuk keperluan analisa kekeruhan air atau larutan. *Sensor* ini merupakan alat pengujian kekeruhan dengan sifat optik akibat dispersi sinar dan dapat dinyatakan sebagai perbandingan cahaya yang dipantulkan terhadap cahaya yang datang. Intensitas cahaya yang dihasilkan *sensor* ini tergantung dari tingkat kekeruhan air, semakin besar intensitas cahaya akan menaikkan nilai hambatan dari *sensor* tersebut. Alat ini banyak digunakan dalam pengolahan air bersih untuk memastikan bahwa air yang akan digunakan memiliki kualitas yang baik dilihat dari tingkat kekeruhannya. Gambar 2.1 menunjukkan bentuk fisik dari *Sensor* Kekeruhan yang digunakan pada alat.

Spesifikasi dari *Turbidity Sensor* adalah sebagai berikut.

1. Tegangan Operasi: 5V DC
2. Operasi Saat Ini: 40mA (maksimum)
3. Waktu Respon: < 500 ms
4. Resistensi Insulasi: 100M (Minimum)
5. Metode Keluaran: *Analog*
6. *Output analog*: 0-4,5 V
7. *Output Digital*: Sinyal tingkat tinggi / rendah (Anda dapat mengatur nilai ambang batas dengan menyesuaikan potensiometer)
8. Suhu Operasional: 5 °C ~ 90 °C
9. Suhu Penyimpanan: -10 °C ~ 90 °C
10. Berat: 30g
11. Dimensi *Adapter*: 38mm * 28mm * 10mm / 1,5inches * 1,1 inches * 0,4 inches



Gambar 2.1 *Sensor Kekeruhan*

2.3 SCADA [3]

SCADA (*Supervisory Control and Data Acquisition*) adalah sistem yang memungkinkan pengguna/ operator untuk melakukan:

1. *Monitoring* (pengawasan)
2. *Controlling* (pengendalian)
3. *Data Acquisition* (pengambilan dan perekaman data)

Ketiga fungsi di atas dapat dipenuhi dengan mewujudkannya dalam bentuk *hardware* maupun *software*. SCADA merupakan sebuah sistem komputer untuk mengumpulkan dan menganalisa data secara *realtime*. Sistem SCADA digunakan untuk memonitor dan mengendalikan pabrik atau alat-alat pada bidang industri seperti telekomunikasi, kendali air dan limbah, energi, minyak, pemurnian gas, serta transportasi. Salah satu komponen penting dalam SCADA adalah HMI.

Istilah HMI adalah alat untuk menjembatani antara manusia (operator) dengan mesin (*Plant*), sehingga operator dapat mengawasi dan mengendalikan *Plant* dengan mudah.

Secara umum fungsi – fungsi utama Sistem SCADA adalah:

1. Mengakuisisi data atau memproses penerimaan data dari peralatan di lapangan
2. Mengonversi data – data dari lapangan ke dalam format standar yang selanjutnya diproses dan dianalisa untuk dilaporkan kepada operator
3. Sebagai *supervisory control* yang memungkinkan operator untuk melakukan pengendalian pada peralatan – peralatan di lapangan
4. Sebagai *alarm* dan *event* untuk menginformasikan kepada operator bila ada perubahan di dalam sistem sebagai bentuk pengawasan
5. *Tagging* yaitu operator dapat meletakkan informasi tertentu pada peralatan
6. *Post Mortem review* yaitu membantu menentukan akibat pada sistem jika terjadi gangguan besar pada jaringan.

Komponen sistem SCADA terdiri dari 3 bagian yaitu pusat kontrol, media telekomunikasi dan RTU. Pusat Kontrol pada sistem SCADA berupa MTU. MTU merupakan komponen utama SCADA yang berupa komputer utama atau *server*. Media komunikasi pada sistem SCADA merupakan media yang menghubungkan antara peralatan untuk melakukan pertukaran informasi. Pada sistem SCADA diperlukan pertukaran informasi antara RTU dengan MTU. Terdapat 3 jenis bahan yang dapat digunakan sebagai media komunikasi yaitu kabel kontrol, radio dan serat optik. RTU adalah mikroprosesor yang bertugas melakukan *scanning*, pengolahan dan penyimpanan data di memori sementara sebelum diminta oleh pusat kontrol dan melakukan aksi atau kendali sesuai permintaan dari pusat kontrol.

2.4 LabVIEW Software [4]

LabVIEW (*Laboratory Virtual Instrument Engineering Workbench*) adalah sebuah bahasa pemrograman *graphical* yang menggunakan simbol-simbol (*icon*) untuk membuat aplikasi. *Visual* atau *virtual instrument* adalah *program* LabVIEW yang menirukan instrumen sebenarnya dalam bentuk simbol-simbol. LabVIEW merupakan *software* khusus yang digunakan untuk pemrosesan dan visualisasi data

dalam bidang akuisisi data, kendali dan instrumentasi, serta otomatisasi industri.

Beberapa kelebihan LabVIEW dibandingkan Bahasa pemrograman lainnya adalah:

1. Bahasa pemrograman LabVIEW jelas dan mudah dipahami karena berbentuk grafis, dengan instruksi berbentuk ikon – ikon yang dihubungkan dengan garis/ kawat untuk menunjukkan aliran data, mirip seperti *Flowchart*
2. Pembuatan *program* mudah, yaitu dengan menarik keluar ikon instruksi yang sudah tersedia di palet dan menghubungkannya dengan kawat ke ikon yang lain. Kawat ini sama seperti variabel pada Bahasa pemrograman teks. Dengan cara ini, LabVIEW menyederhanakan pemrograman
3. Mempersingkat waktu pemrograman dengan gambar yang interaktif dan mudah untuk melakukan perbaikan *program*
4. LabVIEW didesain sebagai Bahasa pemrograman paralel yang mampu menangani beberap instruksi sekaligus dalam waktu bersamaan
5. Bersifat modular sehingga pengguna dapat membuat *program* yang kompleks menjadi sederhana dalam bentuk *SubVi*

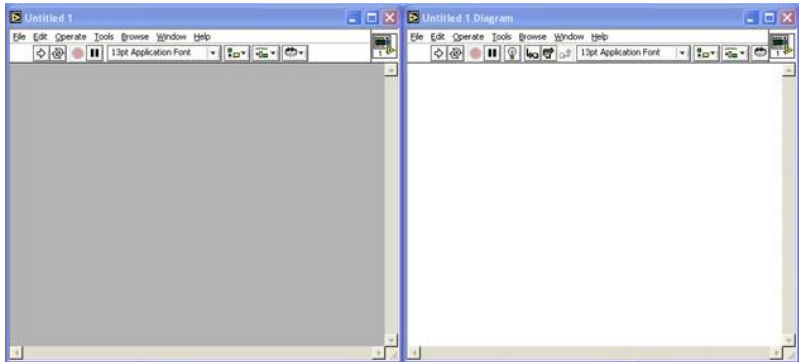
Untuk membuat tampilan *program* aplikasi LabVIEW, digunakan *tools* dan objek. Tampilan ini dikenal dengan istilah *Front Panel*. Lalu ditambahkan kode yang direpresentasikan oleh simbol dari fungsi untuk mengatur objek pada *Front Panel*. *Source code* simbol ini disebut diagram blok . Dalam membuat suatu aplikasi VIs, harus diperhatikan tipe data tiap simbol agar *dataflow* dapat berjalan tanpa kesalahan. Tipe data dari sebuah simbol dapat diketahui dari warna *node* atau warna kabel (*wire*) ketika dihubungkan ke simbol lainnya. Ketika mulai menjalankan aplikasi LabVIEW Terdapat 2 tampilan yang akan selalu digunakan yaitu *Front Panel* dan Diagram blok.

2.4.1 Front Panel

Front panel umumnya berisikan kontrol dan indikator sebagai masukan/keluaran interaktif VIs. Gambar 2.2 menunjukkan *Front Panel* pada LabVIEW. Kontrol adalah instrumen mekanisme masukan yang menyuplai data dari diagram blok, mencakup *knob*, *push button*, *dial*, dan mekanisme masukan lainnya. Sedangkan indikator adalah instrumen mekanisme keluaran yang menampilkan data dari diagram blok, mencakup grafik, LED, *tank*, dan tampilan keluaran lainnya.

2.4.2 Diagram Blok

Diagram Blok adalah jendela tempat menuliskan perintah dan fungsi, berisikan *source code* berupa simbol-simbol, *node* dan garis sebagai *data flow* untuk mengeksekusi *program*, termasuk kode dari *front panel*. Dalam membuat suatu aplikasi VIs, harus diperhatikan tipe data tiap simbol agar *flow data* dapat berjalan tanpa kesalahan. Tipe data dari sebuah simbol dapat diketahui dari warna *node* atau warna kabel (*wire*) ketika dihubungkan ke simbol lainnya



Gambar 2.2 Front Panel Dan Diagram Blok

2.5 Webserver [5]

Webserver merupakan sebuah aplikasi perangkat lunak. Aplikasi *Webserver* dijalankan pada sebuah komputer yang disebut dengan *server*. *Webserver* adalah salah satu jenis perangkat lunak yang menyediakan layanan halaman *website* yang dapat diakses seluruh dunia melalui *internet*. *Webserver* akan melayani permintaan akses halaman *website* dengan bantuan *protocol* komunikasi HTTP. Salah satu contoh *platform* tersebut adalah Apache. Apache adalah aplikasi *webserver* yang tersedia secara gratis dan disebar dengan lisensi *open source*, dengan menggunakan Apache dapat membangun *world wide web* (www).

Sebuah *webserver* dapat diakses dan berkomunikasi dengan pengguna apabila memiliki halaman *interface*. Halaman *interface* suatu *webserver* dibuat menggunakan Bahasa pemrograman HTML. HTML yaitu salah satu Bahasa *scripting* yang dapat menghasilkan halaman *website* sehingga halaman tersebut dapat diakses setiap komputer pengakses (*client*). Dokument HTML merupakan dokumen yang disajikan dalam *browser website*.

Suatu *webserver* memerlukan sebuah *database* untuk menyimpan data, melakukan pengolahan data, menyediakan data *client*, dan melakukan pengamanan data. *Database* manajemen sistem *webserver* yang memiliki kemampuan baik adalah *Oracle* dan *PostgreSQL*. Sedangkan yang umumnya digunakan adalah *MySQL*. *MySQL* merupakan *software* yang bersifat *open source*. *MySQL* digunakan untuk memudahkan dalam *manage database* yang dibuat, baik dalam penambahan tabel, *record* dan *field* maupun menghapus dan mengedit *database* yang ada.

Suatu *database* tidak dapat diakses langsung oleh *client* sehingga diperlukan suatu Bahasa pemrograman yang menjembatani antara *database* dengan *client* yaitu Bahasa pemrograman PHP. Bahasa pemrograman PHP bekerja dalam sebuah *webserver*. *Script* PHP yang dibuat harus tersimpan dalam *webserver* dan dieksekusi atau diproses dalam *server* tersebut. Beberapa keunggulan yang dimiliki *program* PHP:

1. Beberapa *server* seperti Apache, *Microsoft IIS*, *PWS*, *AOLserver*, *phhttpd*, *fhhttpd*, dan *Xitami* mampu menjalankan PHP
2. Tingkat akses PHP lebih cepat serta memiliki tingkat keamanan yang tinggi.
3. Beberapa *database* yang sudah ada baik yang bersifat gratis ataupun komersil sangat mendukung akses PHP salah satu diantaranya *MySQL*
4. PHP mampu berjalan di Linux sebagai *platform* sistem operasi utama bagi PHP.

2.6 Arduino Uno [6]

Arduino Uno adalah *board* berbasis mikrokontroler pada AT-mega328. *Board* ini memiliki 14 *digital input / output* pin (dimana 6 pin dapat digunakan sebagai *output* PWM), 6 *input analog*, 16 MHz *osilator* kristal, koneksi USB, *jack* listrik tombol *reset*. Pin-pin ini berisi semua yang diperlukan untuk mendukung mikrokontroler, hanya terhubung ke komputer dengan kabel USB atau sumber tegangan bisa didapat dari adaptor AC-DC atau baterai untuk menggunakannya. Gambar 2.3 menunjukkan bentuk fisik dari perangkat Arduino Uno.

Kelebihan Arduino dari platform hardware mikrokontroler lain adalah:

1. IDE Arduino merupakan *multiplatform*, yang dapat dijalankan di berbagai sistem operasi, seperti Windows, Macintosh, dan Linux.

2. IDE Arduino dibuat berdasarkan pada IDE *Processing*, yang mempunyai kelebihan dalam hal kesederhanaannya sehingga mudah digunakan.
3. Pemrograman Arduino menggunakan kabel yang terhubung dengan *port USB*, bukan *port serial*. Hal ini sangat berguna karena komputer jaman sekarang jarang sekali yang mempunyai *port serial*.
4. Arduino adalah *hardware* dan *software* yang bersifat *open source*, semua orang dapat mengunduh *software* dan gambar rangkaian Arduino tanpa harus membayar kepada pembuat Arduino.
5. Biaya pembuatan *hardware* cukup murah, sehingga tidak terlalu menghabiskan biaya jika dalam eksperimen nantinya dapat membuat kesalahan yang pada akhirnya menuntut penggantian komponen penyusunnya.
6. Proyek Arduino dikembangkan dalam lingkungan pendidikan, sehingga bagi pemula pun akan lebih cepat dan mudah dalam mempelajarinya.
7. Arduino memiliki banyak pengguna di seluruh dunia, tergabung dalam komunitas di *internet* sehingga siap membantu apabila kita menemui kesulitan dalam mempelajarinya



Gambar 2.3 Arduino Uno

2.6.1 Spesifikasi Arduino

Pada board Arduino terdapat beberapa komponen penyusun antara lain:

1. *Jack USB*
2. *Jack Power*
3. Prosesor
4. *Cip komunikasi / Serial Converter*
5. Kristal 16 MHz

6. Tombol *Reset*
7. LED *Power*
8. LED TX/RX
9. LED (Terhubung ke pin 13)
10. *Pin Power*
11. *Input Analog*
12. *Pin TX dan RX*
13. *Inputs/ Outputs digital*
14. *Ground* dan pin AREF
15. ICSP untuk ATmega328
16. ICSP untuk antarmuka USB

Tabel 2.1 Spesifikasi Arduino Uno

Mikrokontroler	ATmega328
Tegangan Kerja	5 V
<i>Input</i> Tegangan(disarankan)	7 – 12 V
<i>Input</i> Tegangan(batas)	6 – 20 V
<i>Digital I/O</i>	14 pin (6 dapat dijadikan <i>output</i> PWM)
<i>Analog Input</i>	6 Pin
Arus DC per I/O	40 mA
Arus DC untuk 3,3 V	50 mA
Flash <i>Memory</i> 32KB (AT-Mega328)	0,5 KB digunakan untuk <i>boot-loader</i>

Keterangan:

a. *Power*

Arduino uno dapat beroperasi dengan diberi catu daya melalui koneksi USB atau dengan catu daya eksternal, pemilihannya pun dapat dipilih secara otomatis. Arduino dapat beroperasi dengan catu daya eksternal yang bertegangan dari 6 – 20 Volt. Jika diberikan tegangan kurang dari 7 Volt maka Arduino berjalan tidak stabil

b. *Memori*

ATmega328 memiliki memori sebesar 32KB dan juga memiliki memori sebesar 2KB dari SRAM dan 1KB dari EEPROM

c. *Input dan Output*

Masing – masing dari 14 pin *digital* Arduino dapat digunakan sebagai *input* atau *output*, dengan menggunakan fungsi *pinMode()*, *digitalWrite()*, dan *digitalRead()*. Setiap pin dapat

memberikan atau menerima maksimum 40 mA dan memiliki resistor *pull-up* internal (terputus secara *default*) dari 20 – 50 KOhms.

2.7 Relay [7]

Relay adalah saklar elektronik yang dapat membuka atau menutup rangkaian dengan menggunakan kontrol dari rangkaian elektronik lain. Sebuah *relay* tersusun atas kumparan, pegas, saklar (terhubung pada pegas) dan 2 kontak elektronik (*Normally close* dan *Normally open*). Saklar terhubung dengan kontak *Normally Close* saat *relay* tidak aktif atau dapat dikatakan saklar dalam kondisi terbuka. Saklar terhubung dengan kontak *Normally open* saat *relay* aktif atau dapat dikatakan saklar dalam kondisi tertutup. Berdasarkan pada prinsip dasar cara kerjanya, *relay* dapat bekerja karena adanya medan magnet yang digunakan untuk menggerakkan saklar. Saat kumparan diberikan tegangan sebesar tegangan kerja *relay* maka akan timbul medan magnet pada kumparan karena adanya arus yang mengalir pada lilitan kawat. Kumparan yang bersifat sebagai elektromagnet ini kemudian akan menarik saklar dari kontak *NC* (*Normally close*) ke kontak *NO* (*Normally open*). Jika tegangan pada kumparan dimatikan maka medan magnet pada kumparan akan hilang sehingga pegas akan menarik saklar ke kontak *NC* (*Normally close*). Pada Gambar 2.4 merupakan bentuk fisik dari *Relay 4 Channel*.

Relay 4 Channel yang digunakan pada alat ini memiliki spesifikasi sebagai berikut.

1. Menggunakan 4 buah *DI-Relay*
2. Menggunakan tegangan rendah, 5 VDC, sehingga dapat langsung dihubungkan pada sistem mikrokontroler.
3. Tipe *relay* adalah SPDT (*Single Pole Double Throw*): 1 *Common*, 1 *NC* (*Normally Close*), dan 1 *NO* (*Normally Open*).
4. Memiliki daya tahan sampai dengan 10 A.
5. Dapat langsung dihubungkan pada *DI-Smart AVR System*
6. Pin pengendali dapat dihubungkan dengan *pin port* mikrokontroler mana saja, sehingga membuat pemrogram dapat leluasa menentukan pin mikrokontroler yang akan digunakan sebagai pengendali.
7. Dilengkapi rangkaian penggerak (*driver*) *relay* dengan *level* tegangan TTL sehingga dapat langsung dikendalikan oleh mikrokontroler.
8. *Driver* bertipe *active-high* atau kumparan *relay* akan aktif saat pin pengendali diberi logika "1".

9. *Driver* dilengkapi rangkaian peredam GGL induksi sehingga tidak akan membuat *reset* sistem mikrokontroler.

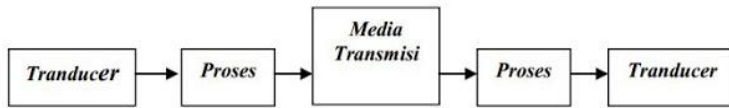


Gambar 2.4 *Relay 4 Channel*

2.8 Telemetri [8]

Telemetri adalah sebuah teknologi pengukuran dilakukan dari jarak jauh dan melaporkan informasi kepada perancang atau operator sistem. Kata Telemetri berasal dari bahasa yunani yaitu *tele* artinya jarak jauh sedangkan *metron* artinya pengukuran. Secara istilah Telemetri diartikan sebagai suatu bidang keteknikan yang memanfaatkan instrumen untuk mengukur panas, radiasi, kecepatan atau *property* lainnya dan mengirimkan data hasil pengukuran ke penerima yang letaknya jauh secara fisik, berada diluar dari jangkauan pengamat atau *user*. Gambar 2.5 merupakan diagram blok Telemetri.

Komunikasi merupakan suatu kata yang dapat diartikan sebagai cara untuk menyampaikan atau menyebarluaskan data dan informasi sedangkan informasi adalah berita, pikiran, pendapat dalam berbagai bentuk. Komunikasi data adalah bagian dari komunikasi yang secara khusus berkenaan dengan transmisi atau pemindahan data dan informasi diantara komputer dan piranti yang lain dalam bentuk *digital* yang dikirim melalui media komunikasi data. Data merupakan informasi yang disajikan oleh isyarat *digital*.



Gambar 2.5 Diagram Blok Telemetri

Telemetri merujuk pada komunikasi nirkabel, tetapi juga dapat merujuk pada data yang dikirimkan melalui media lain, seperti telepon atau jaringan komputer atau melalui sebuah kabel optik. Adapun penjelasan dari diagram blok antara lain:

1. *Tranducer*
Merupakan komponen yang bertugas mengirimkan informasi. Tugas dari komponen ini adalah membangkitkan data atau informasi dan menempatkannya pada media transmisi.
2. *Proses*
Befungsi untuk mengubah informasi yang akan dikirim menjadi bentuk yang sesuai dengan media transmisi yang digunakan.
3. *Media Transmisi*
Merupakan jalur transmisi tunggal atau jaringan transmisi kompleks yang menghubungkan sistem sumber dengan sistem tujuan. Kadang media transmisi juga disebut sebagai pembawa data yang dikirim.
4. *Proses*
Befungsi mengubah informasi yang telah diterima dari pengirim melalui media transmisi. Bagian ini sinyal dari pengirim diterima dari media transmisi.
5. *Tranducer*
Merupakan sistem yang berfungsi untuk menerima sinyal dari sistem transmisi dan menggabungkannya kedalam bentuk tertentu yang dapat ditangkap oleh sistem tujuan.

2.9 *Internet of Things* [9]

Internet of Things atau dikenal juga dengan singkatan IoT, merupakan sebuah konsep yang bertujuan untuk memperluas manfaat dari konektivitas *internet* yang tersambung secara terus-menerus. Adapun kemampuan seperti berbagi data, *remote control*, dan sebagainya, termasuk juga pada benda di dunia nyata. Contohnya bahan pangan, elektronik, koleksi, peralatan apa saja, termasuk benda hidup yang

semuanya tersambung ke jaringan lokal dan global melalui *sensor* yang tertanam dan selalu aktif. Cara Kerja *Internet of Things* yaitu dengan memanfaatkan sebuah argumentasi pemrograman yang dimana tiap perintah argumennya itu menghasilkan sebuah interaksi antara sesama mesin yang terhubung secara otomatis tanpa campur tangan manusia dan dalam jarak berapa pun. *Internet* merupakan penghubung di antara kedua interaksi mesin tersebut, sementara manusia hanya bertugas sebagai pengatur dan pengawas bekerjanya alat tersebut secara langsung.

Tantangan terbesar dalam mengkonfigurasi *Internet of Things* ialah menyusun jaringan komunikasinya sendiri, yang dimana jaringan tersebut sangatlah kompleks, dan memerlukan sistem keamanan yang ketat. Selain itu biaya yang mahal sering menjadi penyebab kegagalan yang berujung pada gagalnya produksi. Konsep IoT ini mengacu pada 3 elemen utama pada arsitektur IoT, yakni:

1. Barang Fisik yang dilengkapi modul IoT,
2. Perangkat Koneksi ke *Internet* seperti Modem dan *Router Wireless*
3. *Cloud Data Center* tempat untuk menyimpan aplikasi beserta *database*.

Dengan prinsip utama dari IoT sebagai sarana yang memudahkan untuk pengawasan dan pengendalian barang fisik maka konsep IoT ini sangat memungkinkan untuk digunakan hampir pada seluruh kegiatan sehari-hari. Gambar 2.6 merupakan ilustrasi sistem kerja dari IoT yang menghubungkan beberapa *device* sekaligus. IoT juga sangat berguna dalam otomatisasi seluruh perangkat yang terhubung ke *internet* dimana konfigurasi otomatisasi tersebut dapat di sesuaikan dengan mudah tanpa harus datang ke lokasi perangkat tersebut. Dan pada wilayah yang tidak mungkin dimasuki manusia, maupun untuk alasan jangkauan terhadap perangkat yang akan di kendalikan tersebut.



Gambar 2.6 IoT Clouds

2.10 Thingspeak [10]

Thingspeak merupakan *Platform* IoT yang dibuat berbasis pada Matlab. Pada *platform* ini *user* dapat mengunggah data *sensor* dari berbagai macam *development board* yang ada. Data yang di *upload* pada *Thingspeak* bisa dibuat sebagai data pribadi ataupun data publik. Data tersebut disajikan dalam bentuk *channel* yang didalamnya terdapat visualisasi yang diolah oleh Matlab. Untuk menggunakan *Thingspeak* saat ini masih tidak dikenakan biaya namun *Sensor* yang digunakan pun masih dibatasi. Ketika mengakses *website Thingspeak* maka akan muncul tampilan seperti pada Gambar 2.7. Fitur dari *ThingSpeak* antara lain:

1. *Open API*
2. Pengumpulan data *realtime*
3. Data geolokasi
4. Pengolahan data
5. Visualisasi data
6. Pesan status perangkat
7. Plugin

ThingSpeak dapat diintegrasikan dengan, Arduino, Raspberry Pi, IoBridge, Aplikasi *Mobile / Website*, Jaringan Sosial dan Analisis Data dengan Matlab.



Gambar 2.7 Halaman Awal Website *Thingspeak*

2.11 Ethernet Shield [11]

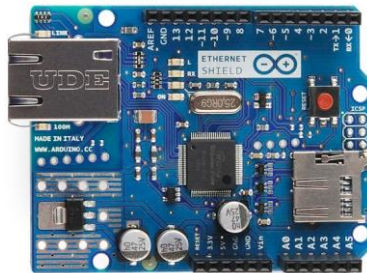
Ethernet Shield menambah kemampuan arduino board agar terhubung ke jaringan komputer. *Ethernet shield* berbasiskan *chip Ethernet Wiznet W5100*. *Ethernet library* digunakan dalam menulis *program* agar arduino board dapat terhubung ke jaringan dengan menggunakan arduino *Ethernet Shield*.

Pada *Ethernet Shield* terdapat sebuah slot *micro-SD*, yang dapat digunakan untuk menyimpan *file* yang dapat diakses melalui jaringan. *Onboard micro-SD card reader* diakses dengan menggunakan *SD li-*

brary. Arduino board berkomunikasi dengan W5100 dan SD card menggunakan bus SPI (*Serial Peripheral Interface*). Komunikasi ini diatur oleh *library* SPI.h dan Ethernet.h. Bus SPI menggunakan pin *digital* 11, 12 dan 13 pada Arduino Uno.

Pin *digital* 10 digunakan untuk memilih W5100 dan pin *digital* 4 digunakan untuk memilih SD card. Pin-pin yang sudah disebutkan sebelumnya tidak dapat digunakan untuk *input /output* umum ketika menggunakan *Ethernet Shield*.

Karena W5100 dan SD card berbagi bus SPI, hanya salah satu yang dapat aktif pada satu waktu. Jika menggunakan kedua perangkat dalam *program* secara bersamaan, hal ini dapat diatasi oleh *library* yang sesuai. Untuk melakukan hal ini pada SD card, set pin 4 sebagai *output* dan menuliskan logika tinggi, sedangkan untuk W5100 yang digunakan adalah pin 10. Pada Gambar 2.8 merupakan bentuk fisik dari *Ethernet Shield*.



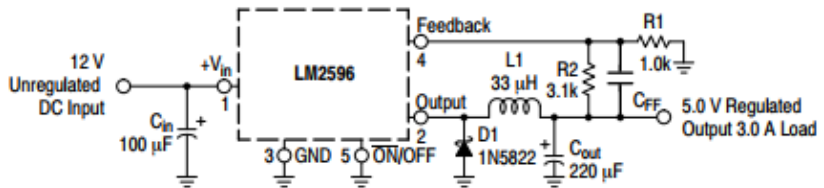
Gambar 2.8 *Ethernet Shield*

2.12 Buck Converter [12]

Buck-converter adalah konverter penurun tegangan khusus yang menerapkan sistem SMPS (*Switching Mode Power Supply*). *Buck converter* merupakan konverter dengan efisiensi yang lebih tinggi jika dibandingkan dengan *power supply* penurun tegangan biasa (sistem linier). Efisiensinya dapat mencapai lebih dari 90%. *Buck converter* yang digunakan pada Tugas Akhir ini adalah tipe LM2596. Bentuk fisik dari *Buck Converter* pada Gambar 2.10

Buck converter memanfaatkan sifat induktor terhadap guncangan listrik berfrekuensi tinggi. Karena itu di dalam sebuah rangkaian *buck converter* selalu terdapat generator sinyal, transistor penguat, dioda, kondensator dan induktor. Konsep dasar rangkaiannya dapat digambarkan pada Gambar 2.9.

Umumnya *buck converter* bekerja dalam “*continuous-mode*” di mana arus dari induktor ketika pelepasan energi senantiasa diupayakan agar tidak mencapai nol sebelum terjadinya proses penyimpanan energi selanjutnya. Untuk mencapai hal ini maka biasanya nilai induktansi dibuat cukup besar dari frekwensi yang dihasilkan oleh generator sinyal.



Gambar 2.9 Rangkaian Buck Converter [18]



Gambar 2.10 Buck Converter

2.13 App Inventor [13]

App Inventor for Android adalah aplikasi *builder* untuk membuat aplikasi yang berjalan di sistem operasi *Android* berbasis *website open source* yang dikelola oleh *Massachusetts Institut of Technology (MIT)*. *App inventor* merupakan aplikasi yang mengubah bahasa kompleks pengkodean berbasis teks menjadi *visual block programming (puzzle)* dengan cara *drag and drop*. *Framework visual programming* yang terkait dengan bahasa pemrograman *drag and drop* dari MIT, secara spesifik merupakan implementasi dari *open block* yang didistribusikan oleh *MIT Scheller Teacher Education program* yg diambil dari riset yang telah dilakukan. Gambar 2.11 merupakan *Icon* dari *website App Inventor*.



Gambar 2.11 Aplikasi *App Inventor Android*

App inventor memudahkan pengguna apabila ingin mendirikan aplikasi *Android* tanpa pemrograman sintaks. *Visual block programming* membantu pengguna untuk memahami *block* dengan mudah walaupun pengguna tidak memiliki dasar cara memprogram namun syarat terpenting untuk dapat menggunakan yaitu pengguna harus memahami terlebih dahulu fungsi logika. Pada aplikasi *Android App Inventor* memiliki kelebihan antara lain:

- 1 Praktis dan tidak rumit
- 2 Tidak memerlukan *coding* hanya membutuhkan logika

Namun, pada aplikasi tersebut juga memiliki kekurangan. Adapun kekurangan dari aplikasi ini adalah

- 1 Komponen yang disediakan tidak lengkap.
- 2 Akses yang terbatas seperti hanya dapat mengakses protokol tertentu yang kompatibel dengan *App Inventor*.

2.14 Motor DC [14]

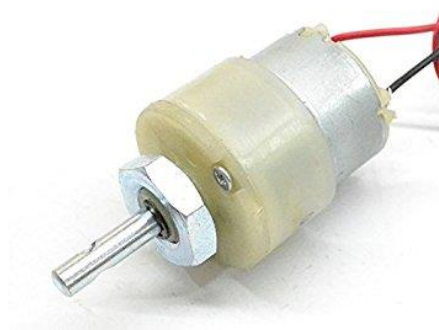
Motor DC adalah motor listrik yang memerlukan *supply* tegangan arus searah pada kumparan medan untuk diubah menjadi energi gerak mekanik. Kumparan medan pada motor dc disebut stator (bagian yang tidak berputar) dan kumparan jangkar disebut rotor (bagian yang berputar). Motor DC digunakan sebagai mengaduk cairan gabungan dari air sungai dengan bahan kimia Tawas (alum). Gambar 2.12 menunjukkan motor DC yang digunakan pada alat.

2.15 Solenoid Valve [15]

Solenoid valve merupakan katup yang dikendalikan dengan arus listrik baik AC maupun DC melalui kumparan / selenoida. *Solenoid valve* ini merupakan elemen kontrol yang paling sering digunakan dalam sistem fluida. Banyak sekali jenis-jenis dari *solenoid valve*, karena *sole-*

noid valve ini di desain sesuai dari kegunaannya. Pada *valve* yang digunakan pada Tugas Akhir ini yaitu *solenoid valve 2/2*. *Solenoid valve* 2 saluran atau yang sering disebut katup kontrol arah 2/2 memiliki 2 jenis menurut cara kerjanya, yaitu *NC* dan *NO*. Jadi fungsinya hanya menutup / membuka saluran karena hanya memiliki 1 lubang *inlet* dan 1 lubang *outlet*. Dimana lubang *inlet* berfungsi sebagai masuknya fluida, lubang *outlet* berfungsi sebagai keluarnya fluida.

Solenoid valve akan bekerja bila kumparan /*coil* mendapatkan tegangan arus listrik yang sesuai dengan tegangan yaitu pada tegangan DC adalah 12/24 VDC sesuai dengan spesifikasi. Gambar 2.13 menunjukkan jenis *Solenoid Valve* yang digunakan.



Gambar 2.12 Motor DC



Gambar 2.13 *Solenoid Valve*

2.16 **Sensor Ketinggian Air** [16]

Sensor ketinggian air merupakan *sensor* yang digunakan untuk mengukur ketinggian air pada sebuah tangki. Adapun *sensor* ketinggian pada Tugas Akhir ini menggunakan Potensiometer *Ten Turn* seperti pada Gambar 2.14. Potensio *Ten Turn* merupakan *sensor* yang memiliki *output* berupa tegangan. Tegangan yang terukur akan dikonversikan pada ketinggian tangki sehingga didapat ketinggian berupa konversi tegangan ke *centimeter* yang terhubung pada pin arduino A0 dan pada pin A1.



Gambar 2.14 *Sensor Ketinggian Air*

-----Halaman ini sengaja dikosongkan-----

BAB III

PERANCANGAN SISTEM

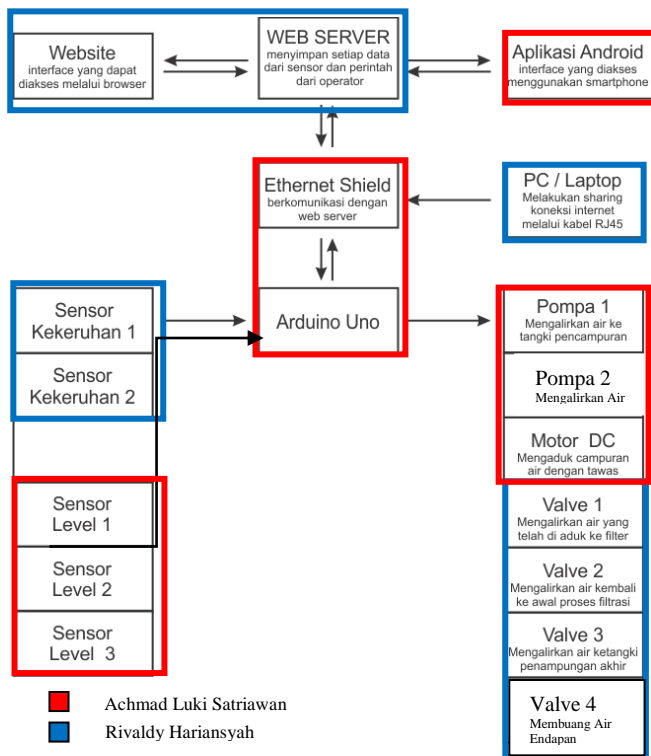
Pada bab ini membahas perancangan sistem dan alat yang akan dibuat. Dimana akan dibagi menjadi dua bagian, untuk Rivaldy Hariansyah bertugas untuk merancang dan membuat *hardware* tangki untuk membaca kekeruhan air dengan *turbidity sensor* serta merancang dan membuat *software* berupa *webserver*, *program* pembacaan kekeruhan air, HMI LabVIEW dan menggabungkan *program* menjadi *program* operasional alat. Untuk Achmad Luki Satriawan bertugas untuk merancang dan membuat *hardware* yang berhubungan dengan tangki penampungan air sungai untuk mengukur ketinggian dari air yang ditampung, mengatur penempatan *input* dan *output* modul arduino membuat aplikasi *Android* menggunakan *App Inventor*, *software* pengukuran ketinggian tangki, pengiriman data dengan *Ethernet Shield* dan membuat *Channel Server* di *Thingspeak*.

3.1 Diagram Fungsional Alat

Perancangan Tugas Akhir secara keseluruhan adalah untuk menjaga ketersediaan air pada proses penjernihan air. Suplai utama listrik berasal dari PLN. Saat suplai PLN aktif, maka akan mengaktifkan *relay* 12 Volt pada pompa 1 yang terhubung ke modul arduino untuk mempompa air sungai dari sumber air untuk dipompa ke tangki penampungan awal. Gambar 3.1 merupakan diagram fungsional alat. Pada saat proses pengisian air pada tangki penampungan awal, dengan waktu yang bersamaan *sensor* kekeruhan akan mendeteksi tingkat kekeruhan air yang mengalir pada pipa. Pada *Sensor level* akan mendeteksi ketinggian air yang masuk pada tangki penampungan awal. Ketika tinggi air pada tangki penampungan awal telah mencapai nilai 25 Cm maka pompa akan berhenti mengalirkan air. Terdapat tombol untuk bahan kimia tawas. Tombol ditekan apabila bahan kimia tawas telah dicampurkan pada tangki penampungan awal dan disaat menekan tombol maka akan mengirim perintah ke arduino bahwa tawas telah diberikan kemudian motor DC akan mengaduk. Setelah pengadukan selama 10 detik motor akan berhenti mengaduk. *Valve 1* akan membuka dan mengalirkan air ke filter.

Pada proses filtrasi air akan melalui beberapa tahap penyaringan yaitu melalui Batu Zeolit, arang, pasir dan kapas. Diakhir proses air dicek kualitasnya menggunakan *sensor* kekeruhan 2. Apabila hasil pem-

bacaan *sensor* kekeruhan 2 air dianggap masih keruh maka *valve* 2 akan terbuka dan mengalirkan air kembali ke awal proses filtrasi tetapi apabila air dianggap jernih maka *valve* 3 akan terbuka dan mengalirkan air ke tangki hasil akhir. Pada tangki penampungan, filter dan tangki penampungan hasil akhir terdapat *sensor level* yang berfungsi untuk mengetahui tingkat ketinggian air. Hasil pengukuran *sensor level* dan kekeruhan akan dikirimkan ke *webserver* menggunakan Arduino yang telah terkoneksi dengan *internet*. Apabila salah satu atau ketiga wadah ini mengalami kelebihan daya tampung (*overload*) maka akan melakukan aksi untuk mengurangi ketinggian air yaitu membuka *valve* 2 dan mematikan pompa 2. Tingkat kekeruhan pada setiap proses akan dibaca oleh *sensor* kekeruhan dan dikirim ke *webserver* agar setiap proses dapat tercatat secara *realtime*.



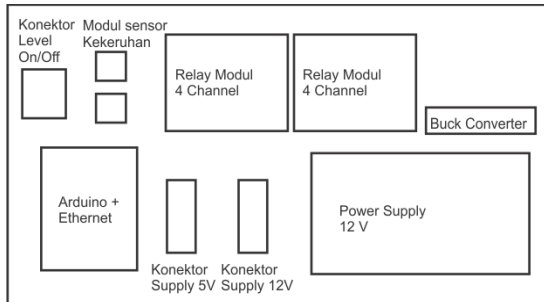
Gambar 3.1 Diagram Fungsional *Prototype* Tugas Akhir

Penjelasan Diagram Fungsional:

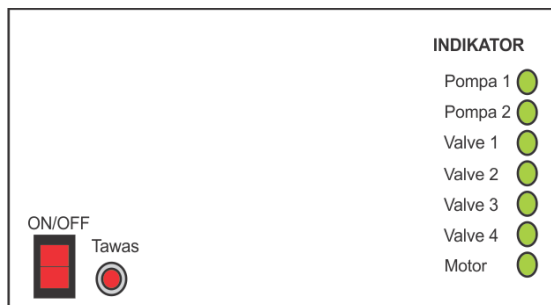
- 1 *Web Server* : *Web Server* berfungsi sebagai tempat penyimpanan data. Data pada *web server* dapat ditambah atau diakses melalui suatu alat / *device* yang terkoneksi dengan jaringan *internet*.
- 2 *Website* : *Website* berisi halaman yang hanya dapat diakses oleh operator. Operator dapat memantau setiap perubahan nilai *sensor* melalui *website* secara *realtime* dan dapat mengubah kondisi dari *actuator*. Untuk mengakses halaman *website* diperlukan suatu aplikasi *browser* di *PC/Android*
- 3 Aplikasi *Android* : Aplikasi *Android* ini memiliki fungsi untuk memantau kondisi *sensor* dan *actuator*. Aplikasi ini dapat diinstall pada *smartphone Android*.
- 4 *Ethernet Shield* : *Ethernet Shield* membantu Arduino dalam mengakses *internet* menggunakan *protocol TCP/IP* karena Didalam *Ethernet Shield* terdapat *MAC Address* dan *IP* untuk berkomunikasi dalam suatu jaringan.
- 5 *PC/ Laptop* : *PC / laptop* menggantikan peran *Router* untuk *sharing* jaringan *internet* ke *Ethernet Shield*.
- 6 *Arduino Uno* : Digunakan untuk mengendalikan aktuatur dan mengakuisisi data dari semua *sensor*
- 7 *Sensor Kekeruhan 1* : Membaca tingkat kekeruhan air sebelum dilakukan proses penjernihan air
- 8 *Sensor Kekeruhan 2* : Membaca tingkat kekeruhan air setelah melalui proses filtrasi dan Sedimentasi
- 9 *Sensor Level 1* : Membaca ketinggian air pada tangki pengadukan
- 10 *Sensor Level 2* : Membaca ketinggian air pada tangki filter
- 11 *Sensor Level 3* : Membaca ketinggian air pada tangki hasil akhir
- 12 Pompa 1 : Mengalirkan air ke tangki penampungan awal
- 13 Pompa 2: Mengalirkan air ke tangki filtrasi
- 14 Motor DC : Mengaduk campuran Air dengan Tawas
- 15 *Valve 1* : Mengalirkan air yang telah diaduk ke filter
- 16 *Valve 2* : Mengalirkan air kembali ke awal filtrasi
- 17 *Valve 3* : Mengalirkan air ke tangki penampungan akhir
- 18 *Valve 4* : Mengalirkan endapan dari tangki penampungan awal
- 19 Rangkaian LED : Indikator Tangki apabila mengalami *overload*

3.2 Perencanaan dan Pembuatan *Hardware*

Perancangan *hardware* pada Tugas Akhir ini, terdiri dari perancangan *sensor* ketinggian, perancangan *sensor* kekeruhan, perancangan sistem *input* dan *output* Arduino. Tampilan desain perancangan *hardware* seperti terlihat pada Gambar 3.2 dan Gambar 3.3



Gambar 3.2 Perancangan *Hardware* Tampak Dalam



Gambar 3.3 Perancangan *Hardware* Tampak Luar

Tata Letak dari komponen-komponen sistem kontrol ini dibuat seperti demikian untuk mempermudah pengoperasiannya.

3.2.1 Perancangan Sistem *Input/ Output* Modul Arduino

Pada Alat Tugas Akhir tentang *Telemetry Level* serta proses penjernihan air berbasis *IoT*, perancangan Sistem *Input / Output* Modul Arduino merupakan tanggung jawab Achmad Luki Satriawan. Alat ini menggunakan modul Arduino Uno sebagai *controller*. Arduino disini bertindak sebagai *controller* yang bertugas mengontrol kerja dan membaca status atau besaran dari komponen-komponen yang ada pada alat ini dan mengirimkan ke *server* menggunakan bantuan modul *Ethernet*. Komponen-komponen yang terhubung dengan modul Arduino yaitu

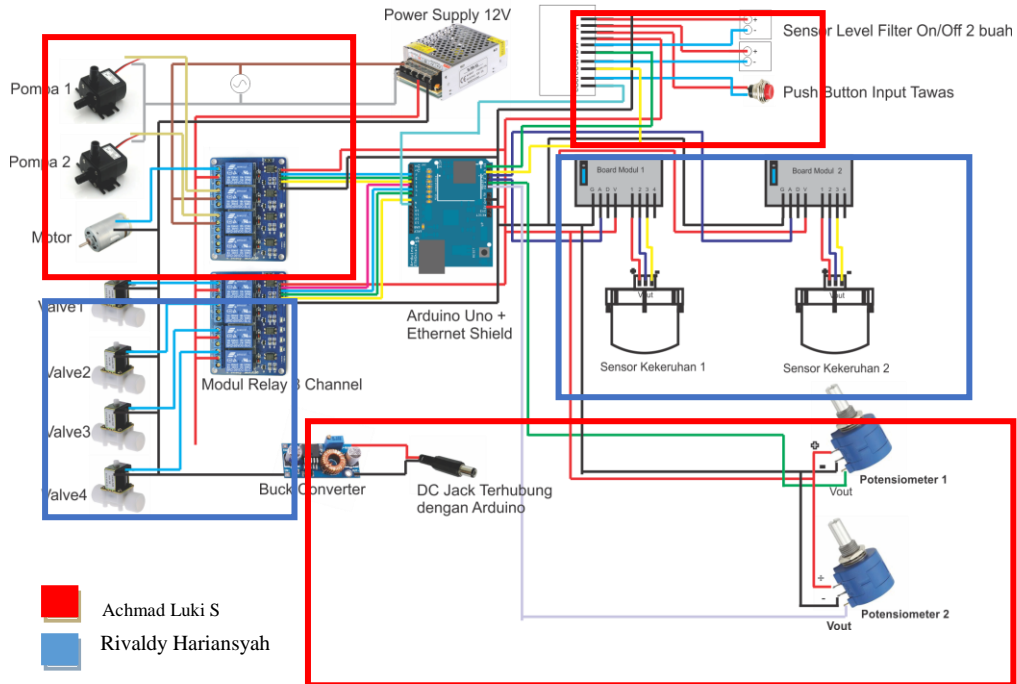
sensor kekeruhan, *sensor* ketinggian, modul *relay* 5V 8 *channel*, *push button*, On/Off *level* dan modul *ethernet*. Masing-masing dari komponen tersebut terhubung dengan pin *digital* maupun *analog* pada modul Arduino. Penghubungan masing-masing komponen tersebut berhubungan dengan pembuatan *program* pada *software* Modul Arduino IDE nantinya. Skematik dari rangkaian keseluruhan komponen yang tersambung dengan Modul Arduino ditunjukkan pada Gambar 3.4. Perincian sambungan Modul Arduino sebagai berikut:

1. Pin 2 hingga 8 dihubungkan dengan LED 1 hingga 7 sebagai *indicator* kondisi dari *actuator* Pompa 1, Pompa 2, *Valve 1*, *Valve 2*, *Valve 3*, *Valve 4* dan Motor dan dihubungkan juga dengan *input data Relay 8 Channel*
2. *Push Button* dihubungkan dengan pin 9
3. *Sensor level* air 1 dan 2 dihubungkan dengan pin A0 dan A1
4. *Sensor* kekeruhan 1 dan 2 dihubungkan dengan pin A2 dan A3
5. *Sensor level* On/Off 1 dan 2 dihubungkan dengan pin A4 dan A5 tetapi pin tersebut bertindak sebagai *input digital*
6. Pin 10 hingga 13 terhubung dengan *ethernet shield*

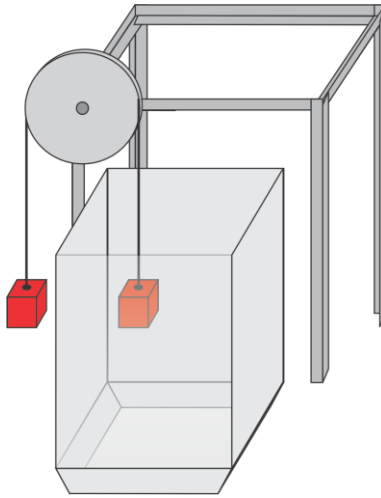
Pembagian tugas pada perancangan keseluruhan alat dapat dilihat Pada Gambar 3.4. Persegi empat warna merah merupakan perancangan yang dilakukan oleh Achmad Luki Satriawan dan pada persegi empat berwarna biru merupakan perancangan yang dilakukan oleh Rivaldy Hariansyah.

3.2.2 Perancangan *Sensor* Ketinggian

Perancangan *sensor* ketinggian merupakan tanggung jawab Achmad Luki Satriawan. *Sensor* ketinggian digunakan untuk membaca tingkat ketinggian air pada tangki penampungan awal dan tangki akhir. *Sensor* ketinggian air yang digunakan adalah Potensiometer *Ten Turn*. Pada *sensor* ini terdapat 3 pin terhubung dengan *Vcc*, *Vout* dan *Ground*. *Sensor* ketinggian yang digunakan sebanyak 2 buah sehingga masing - masing Pin *Vout* terhubung dengan pin A0 dan A1 pada Modul Arduino Uno. Perancangan *sensor* ketinggian seperti pada Gambar 3.5



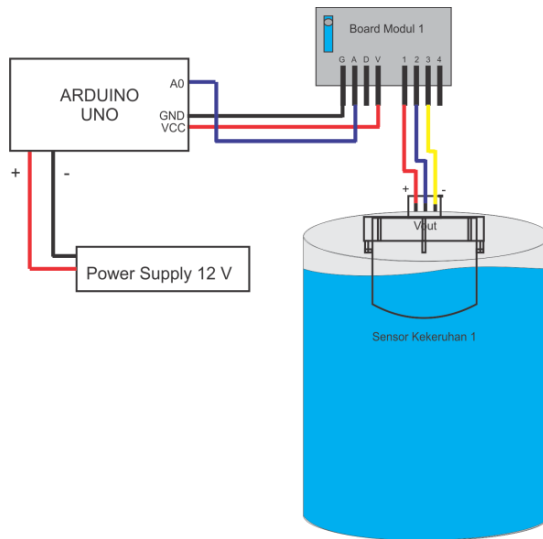
Gambar 3.4 Perancangan I/O Arduino



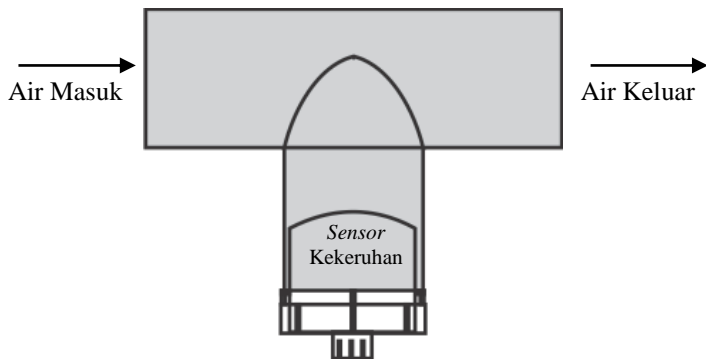
Gambar 3.5 Perancangan *Sensor Ketinggian*

3.2.3 Perancangan *Sensor Kekeruhan*

Perancangan *sensor* kekeruhan merupakan tanggung jawab Rivaldy Hariansyah. *Sensor* kekeruhan digunakan untuk membaca tingkat kekeruhan air yang masuk ke alat dan air hasil proses filtrasi. *Sensor* kekeruhan yang digunakan adalah *turbidity sensor* tipe SEN0189. Pada *sensor* ini terdapat *board* modul yang memiliki 8 buah pin. 3 pin terhubung dengan *Vcc*, *Vout* dan *Ground*. *Sensor* kekeruhan yang digunakan sebanyak 2 buah sehingga masing - masing Pin *Vout* terhubung dengan pin A2 dan A3 pada Modul Arduino Uno . Perancangan elektrik *sensor* kekeruhan seperti pada Gambar 3.6 dan Gambar 3.7 merupakan bentuk fisik pipa air yang digunakan untuk tempat *Sensor* kekeruhan.



Gambar 3.6 Perancangan *Sensor* Kekeruhan

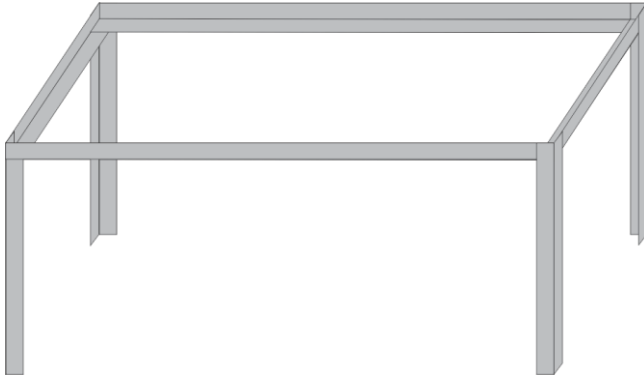


Gambar 3.7 Bentuk Fisik Wadah *Sensor* Kekeruhan

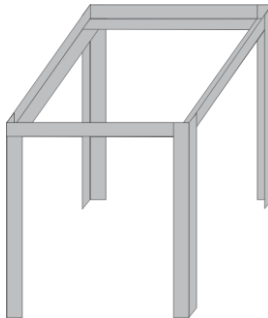
3.2.4 Perancangan Besi Penyangga

Perancangan besi penyangga merupakan tanggung jawab Achmad Luki Satriawan. Besi penyangga berupa sebuah kotak yang tersusun dari besi siku yang disusun sebagai penyangga untuk tempat peralatan dan rangkaian yang berhubungan dengan Tugas Akhir. Perancangan besi penyangga di desain dengan bentuk memanjang yang telah disesuaikan dengan letak dari masing-masing alat. Desain dari besi penyangga filter

dan tangki ditunjukkan pada Gambar 3.8 dan desain besi penyangga kontroler ditunjukkan pada Gambar 3.9



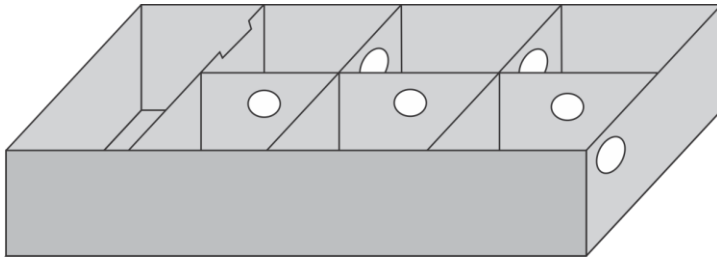
Gambar 3.8 Perancangan Penyangga Akuarium



Gambar 3.9 Perancangan Penyangga Panel Kontrol

3.2.5 Perancangan *Hardware* Aquarium

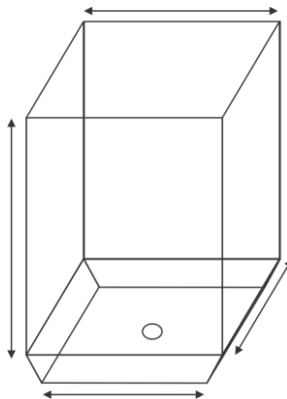
Perancangan Aquarium merupakan tanggung jawab Rivaldy Hari-ansyah. Pada perancangan aquarium yang digunakan pada Tugas Akhir ini merupakan aquarium tipe *continous process* dimana air yang mengalir akan terus diproses. Aquarium ini merupakan peminjaman dari Lab Pengolahan Bahan Kimia D3 Teknik Kimia ITS yang di pinjamkan untuk keperluan penelitian Tugas Akhir. Gambar 3.10 merupakan desain dari *hardware* filter



Gambar 3.10 Perancangan Mekanik Aquarium Untuk Proses Filtrasi

3.2.6 Perancangan Tangki Pengadukan Dan Tangki Final

Perancangan Tangki Pengadukan dan Tangki Final merupakan tanggung jawab Rivaldy Hariansyah. tangki pengadukan dan tangki final didesain dengan ukuran $P=25$ cm dan $L=22$ cm. Desain tangki seperti pada Gambar 3.11. Adapun kapasitas daya tampung pada tangki ini memiliki kapasitas sebesar 12 Liter. Pada perancangan tangki pengadukan dan tangki final dibuat dengan bahan *acrylic*. Penggunaan bahan *acrylic* pada tangki memiliki kemudahan karena bahan yang dipakai ringan dan mudah dibentuk.

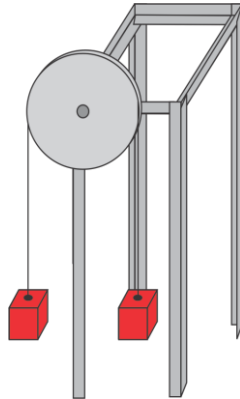


Gambar 3.11 Perancangan Tangki

3.2.7 Perancangan Sensor Ketinggian Air

Perancangan *sensor* ketinggian air merupakan tanggung jawab Achmad Luki Satriawan. *Sensor* yang digunakan yaitu *sensor* potensiometer yang memiliki *output* berupa tegangan. Untuk dapat dilakukan pengukuran air pada tangki salah satu dari kotak dimasukkan pada tangki. Ketika air mengisi tangki maka kotak perlahan ikut naik

dan memicu potensiometer untuk berputar sehingga tegangan akan naik. Perancangan mekanik *Sensor* ketinggian seperti pada Gambar 3.12



Gambar 3.12 Perancangan Mekanik *Sensor* Ketinggian Air

3.3 Perancangan *Software*

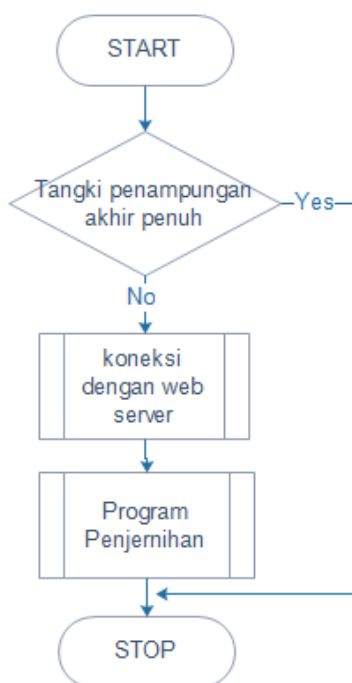
Setelah melakukan perancangan *hardware*, selanjutnya diperlukan sebuah *software* yang mampu mengelola, mengontrol peralatan pendukung lainnya dan berkomunikasi dengan *server* untuk mengirim dan menerima data. *Software program* berisi perintah yang dapat dieksekusi oleh modul Arduino sehingga sistem dapat bekerja sesuai dengan alur dan tujuan yang direncanakan. Perancangan perangkat lunak (*software*) yang digunakan yaitu dengan menggunakan *program* Arduino IDE untuk pembacaan *sensor* kekeruhan, *sensor* ketinggian, dan pengiriman – penerimaan data dari *server*. sebagai *Human Machine Interface* antara operator dengan alat digunakan *software* NI Labview 2015 untuk membuat aplikasi berbasis Windows, *App inventor* membuat aplikasi *Android*, dan Notepad+ untuk membuat halaman *website* sebagai tampilan *server*.

3.3.1 Perancangan Sistem Secara Keseluruhan

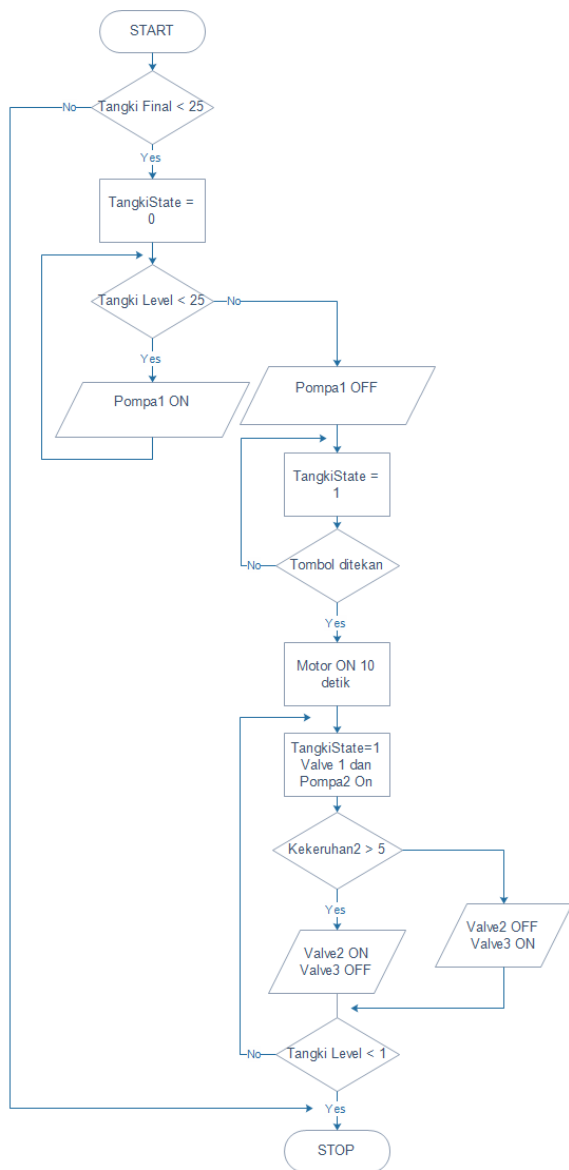
Perancangan sistem secara keseluruhan merupakan tanggung jawab Rivaldy Hariansyah. Secara umum alat ini berjalan dimulai dari koneksi dengan *webserver*, dilanjutkan dengan *program* penjernihan air. Dua *program* ini akan berjalan secara berulang-ulang (*looping*) dengan waktu yang relatif cepat untuk menjaga agar data yang dikirim ke *server* tidak memiliki *delay* lebih dari 30 detik. *Flowchart* dari perancangan sistem keseluruhan dapat dilihat pada Gambar 3.13. *Flowchart program*

penjernihan air dapat dilihat pada Gambar 3.14. Secara keseluruhan sistem pada alat ini dapat ditulis melalui algoritma berikut.

- 1 Kerja alat ini dimulai saat ketinggian air pada tangki pengadukan berada di ketinggian 0 Cm, maka pompa air 1 akan aktif dan mengalirkan air kedalam tangki pengadukan. Pada kondisi ini tangki akan masuk ke dalam *state* 0
- 2 *Sensor* ketinggian akan memantau kondisi ketinggian air, apabila mencapai nilai 25 Cm pompa air akan otomatis mati. *State* tangki akan bernilai 1
- 3 Apabila tangki dalam kondisi *state* 1 maka tidak akan terjadi aktivitas dari alat selain mengirim data ke *server* hingga tawas dicampurkan pada tangki penampungan awal.
- 4 *Push button* akan ditekan ketika tawas telah dicampurkan. *State* tangki berubah menjadi 2 dan motor akan mengaduk selama 10 detik.
- 5 Setelah motor mengaduk selama 10 detik. *Valve* 1 akan aktif dan mengalirkan air ke pompa 2.
- 6 Pompa 2 akan otomatis aktif dan mengalirkan air ke akuarium filter.
- 7 Ketika akuarium dalam kondisi penuh (*sensor* on/off pertama bernilai 1) maka pompa 2 akan berhenti dan *Valve* 1 akan menutup.
- 8 Proses ini akan terus berlangsung hingga ketinggian air di tangki penampungan awal bernilai 0 Cm. Apabila ketinggian tangki bernilai tersebut maka *state* kembali bernilai 0 dan pompa 1 akan aktif kembali
- 9 *Valve* 2 dan *Valve* 3 akan bekerja berkebalikan. Apabila *sensor* kekeruhan menunjukkan sama atau kurang dari 5 NTU maka *Valve* 2 akan terbuka dan *Valve* 3 akan menutup tetapi jika menunjukkan lebih dari 5 NTU
- 10 Disetiap akhir satu *looping program* akan dilakukan proses pengiriman data ke *server*.



Gambar 3.13 *Flowchart* Alat Secara Umum

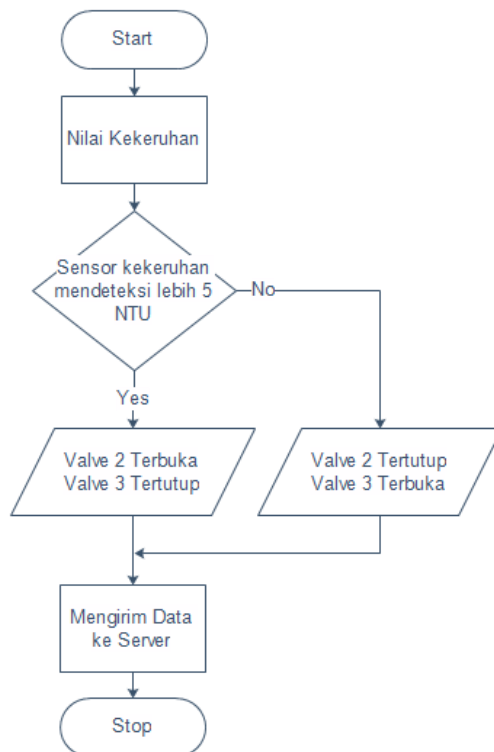


Gambar 3.14 Flowchart Proses Penjernihan Air

3.3.2 Perancangan *Software Sensor Kekeruhan*

Perancangan *software sensor* kekeruhan merupakan tanggung jawab Rivaldy Hariansyah. *Software sensor* kekeruhan dapat digambarkan dengan algoritma dan sebuah *flowchart* seperti pada Gambar 3.15. Algoritma dari sistem *sensor* kekeruhan air adalah sebagai berikut.

- 1 Membaca nilai tegangan yang terdeteksi oleh *sensor*
- 2 Sebelum masuk ke Modul Arduino, nilai tegangan tersebut dikonversi menjadi ADC
- 3 Nilai pembacaan ADC tersebut dikonversi ke satuan NTU dan dikirimkan ke *server*

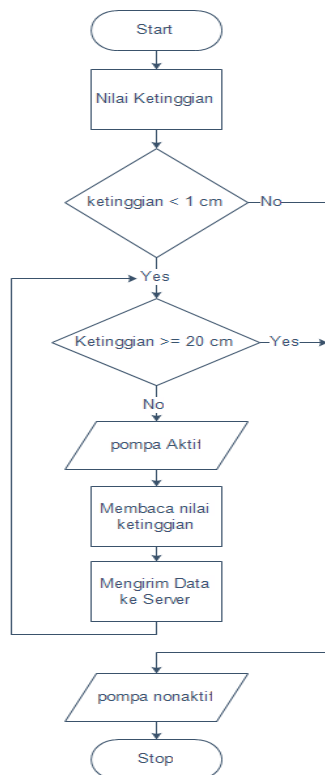


Gambar 3.15 *Flowchart Sensor Kekeruhan*

3.3.3 Perancangan *Software Sensor Ketinggian*

Perancangan *software sensor* ketinggian merupakan tanggung jawab Achmad Luki Satriawan. Perancangan *software* dapat digambarkan dengan algoritma dan sebuah *flowchart* seperti pada Gambar 3.16. Algoritma dari sistem *sensor* ketinggian air adalah sebagai berikut.

- Membaca nilai tegangan yang terdeteksi oleh *sensor*
- Sebelum masuk ke Modul Arduino, nilai tegangan tersebut dikonversi menjadi ADC
- Nilai pembacaan ADC tersebut dikonversi ke satuan *centimeter* dan dikirimkan ke *server*



Gambar 3.16 *Flowchart Sensor Ketinggian*

3.3.4 Perancangan Software Ethernet Shield

Perancangan *software Ethernet Shield* merupakan tanggung jawab Achmad Luki Satriawan. *Software Ethernet Shield* dapat digambarkan dengan algoritma dan sebuah *flowchart* seperti pada Gambar 3.17. Algoritma dari sistem *Ethernet Shield* adalah sebagai berikut.

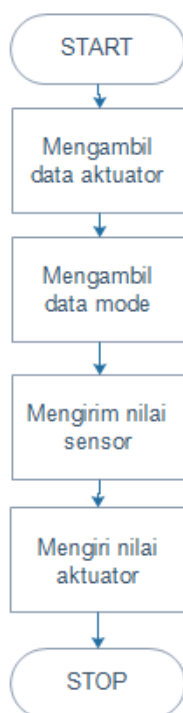
- a. Mendeklarasikan *IP address* dan alamat *server*
- b. Mendeklarasikan bahwa Arduino berperan sebagai *Client*
- c. Memulai koneksi dengan *server* menggunakan perintah *Ethernet.begin*
- d. Jika telah berhasil terkoneksi melakukan pengiriman data dengan perintah *client. Connect*
- e. Merangkai semua data menjadi 1 variabel lalu mengirimkan menggunakan perintah *client.print*
- f. Menunggu respon dari *server*. jika terdapat respon maka menyimpan data ke variabel. Jika tidak ada respon dilanjutkan ke tahap selanjutnya.
- g. Memutuskan koneksi dengan perintah *client stop*

3.3.5 Perancangan Software HMI LabVIEW

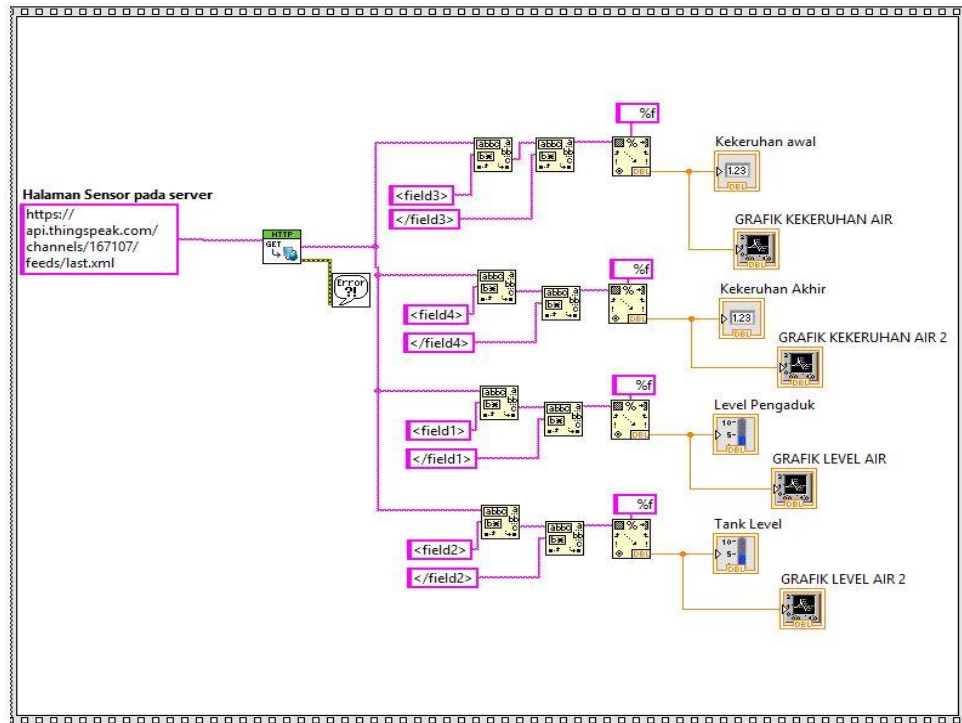
Perancangan *software HMI LabVIEW* merupakan tanggung jawab Rivaldy Hariansyah. Perancangan HMI menggunakan *software NI LabVIEW 2015*. Kerja HMI LabVIEW dapat digambarkan dengan algoritma dan sebuah *flowchart* seperti pada Gambar 3.23. Algoritma dari sistem HMI LabVIEW adalah sebagai berikut.

- a. Mengakses alamat *server* untuk kondisi *sensor*
- b. Data respon dari *server* disimpan dalam variabel masing – masing *sensor* lalu ditampilkan pada *textbox* dan *graph*
- c. Mengakses alamat *server* untuk kondisi aktuator
- d. Data respon dari *server* disimpan dalam variabel masing – masing aktuator lalu ditampilkan pada LED
- e. Apabila pengguna akan melakukan perubahan pada kondisi aktuator, lakukan perubahan pada *button* di Tab konfigurasi. Gambar tab konfigurasi pada Gambar 3.20
- f. Setelah dilakukan perubahan apabila tombol *submit* ditekan maka akan mengakses alamat *server* untuk kondisi aktuator dan mengirim data yang baru. Diagram blok dari proses ini sesuai Gambar 3.21

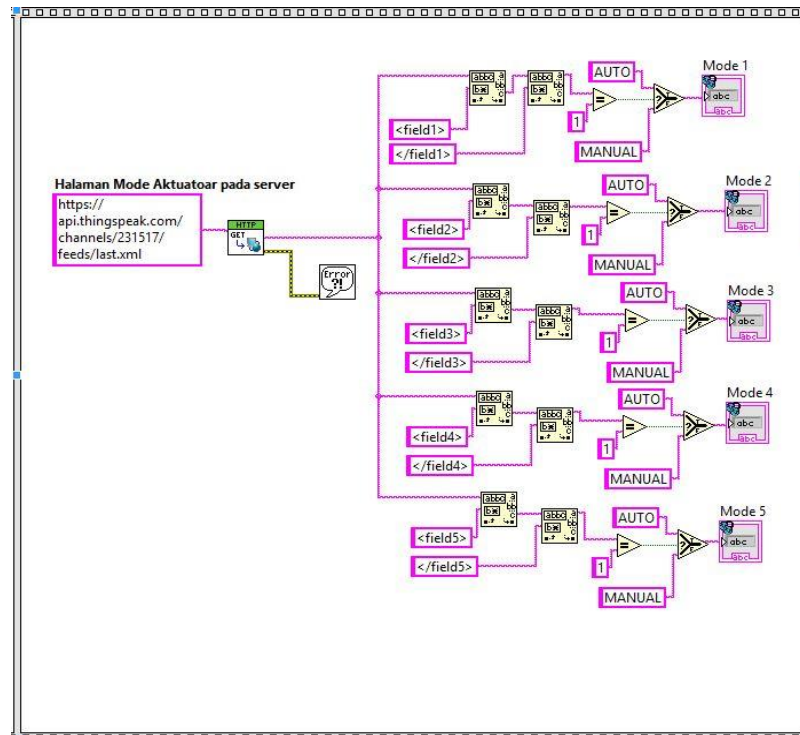
Pada Gambar 3.18, Gambar 3.19 dan Gambar 3.20 menunjukkan diagram blok LabVIEW yang berfungsi untuk mengambil data dari *server* dan mengelola data respon untuk ditampilkan sebagai *indicator*



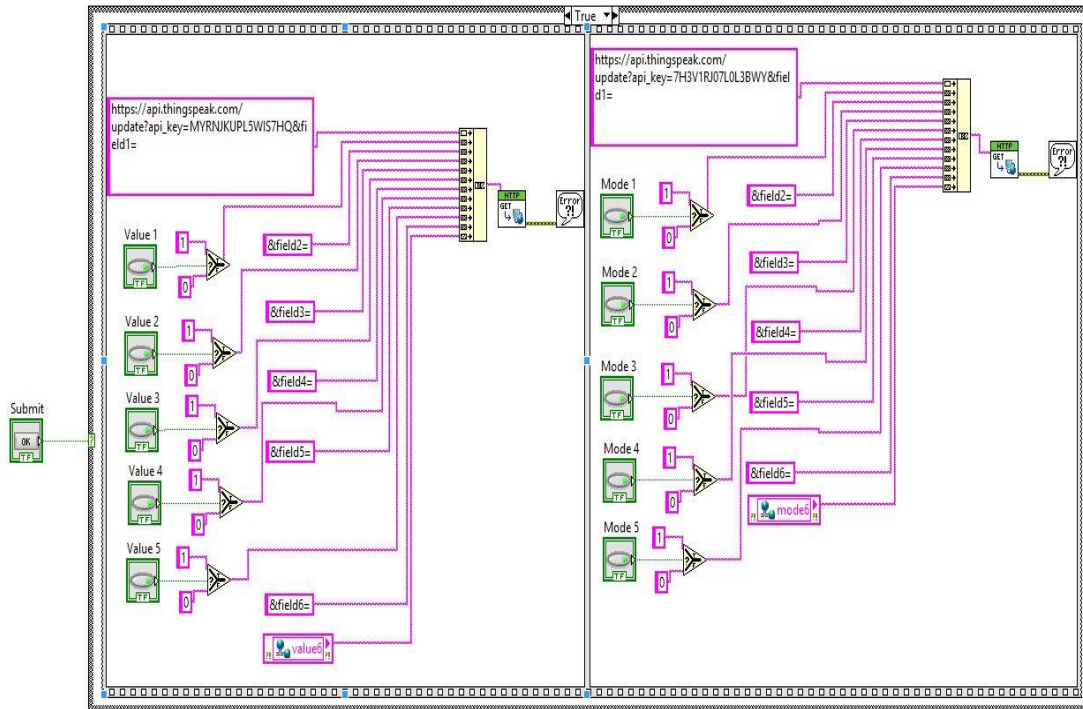
Gambar 3.17 *Flowchart Kerja Ethernet Shield*



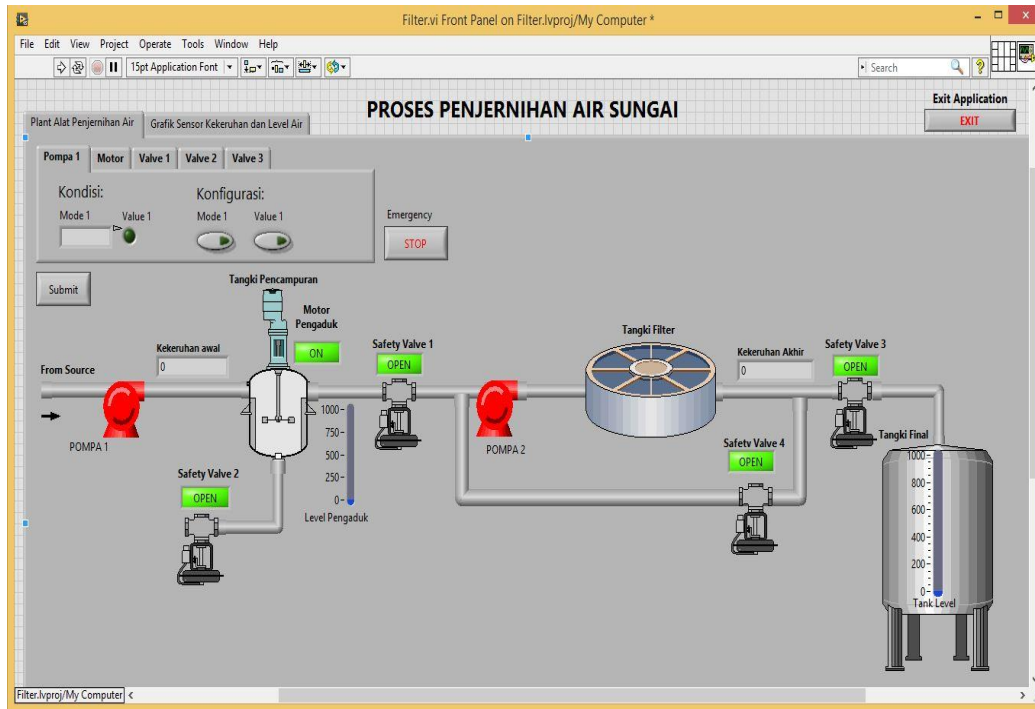
Gambar 3.18 Block Diagram Mengakses Data Sensor



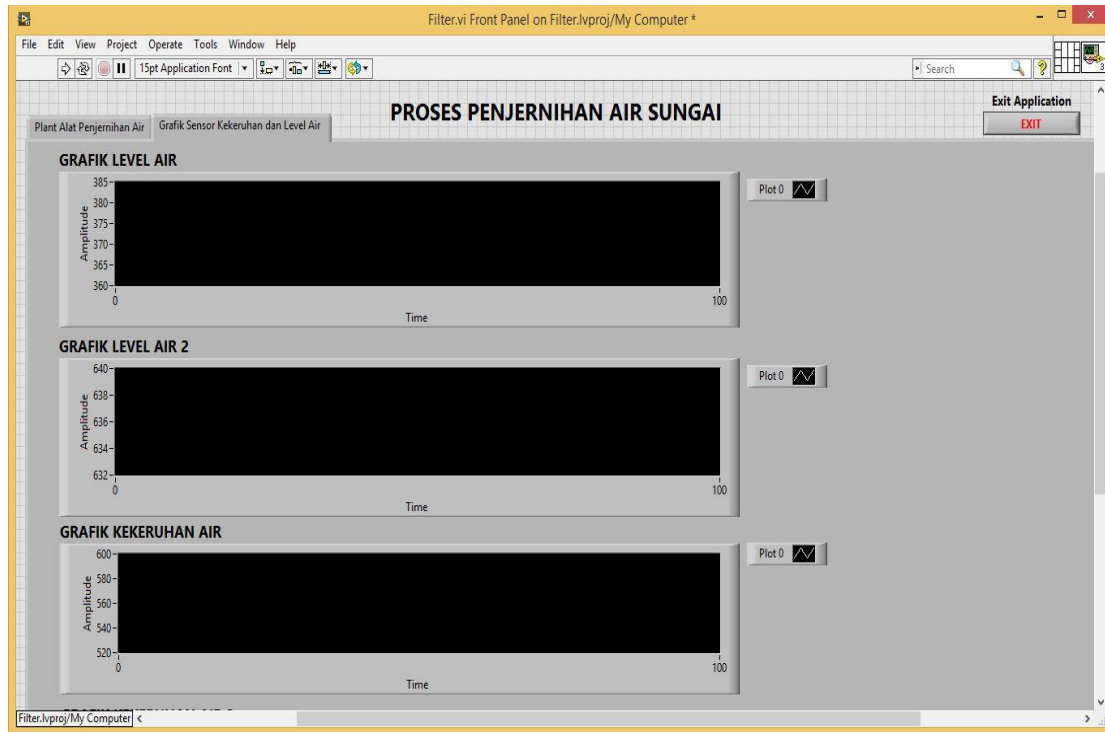
Gambar 3.19 Block Diagram Mengakses Data Aktuator



Gambar 3.20 Block Diagram Mengirim Data Aktuator

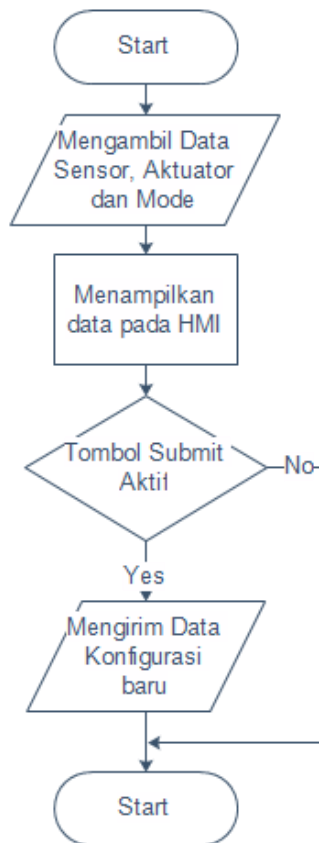


Gambar 3.21 HMI LabVIEW Pengaturan Alat



Gambar 3.22 HMI LabVIEW Grafik *Sensor*

Pada Gambar 3.21 terdapat tampilan 2 buah pompa air dan 4 buah *valve* yang akan berubah warna menjadi merah ketika dalam kondisi tidak aktif / tertutup (tidak mengalirkan air) dan akan berwarna hijau apabila aktif / mengalirkan air. Pada Tab di pojok kiri atas digunakan untuk mengubah konfigurasi dari aktuator. Pengguna harus memasukkan perubahan yang diinginkan pada tiap aktuator sebelum menekan tombol *Submit* dengan cara mengganti nilai *mode* dan *value*. Pada Gambar 3.22 terdapat 3 buah grafik untuk masing – masing *sensor*. Setiap perubahan akan tercatat pada grafik tersebut sejak *program* dimulai hingga *program* berhenti.

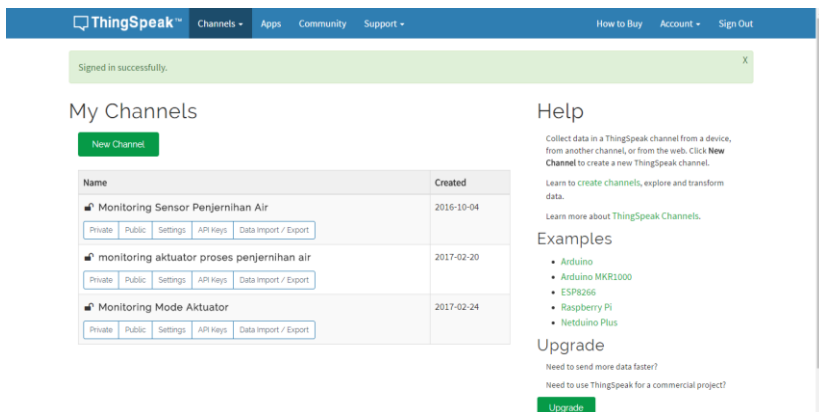


Gambar 3.23 Flowchart LabVIEW

3.3.6 Perancangan Software *Thingspeak*

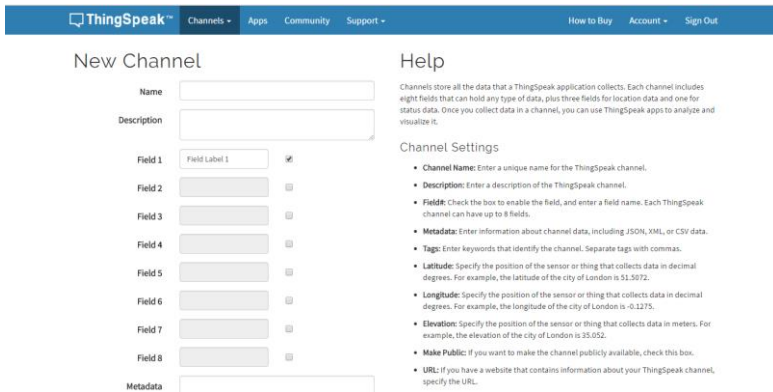
Perancangan *Software Thingspeak* merupakan tanggung jawab Achmad Luki Satriawan. Perancangan *software Thingspeak* menggunakan layanan *website Thingspeak.com* versi 2017. Sebelum dapat mengunggah data hasil pengukuran *sensor* dan kondisi aktuator yang ada di *Arduino*, terlebih dahulu dibuat akun ke *server* yang mempunyai kemampuan menerima data *String*. Pada alat ini digunakan *server thingspeak.com* untuk menerima dan mengirim data *string* dari *Arduino* dengan pertimbangan bahwa dalam situs tersebut dapat memuat banyak data dan mempunyai fasilitas penyimpanan yang cukup banyak. Selain itu, pada *thingspeak.com* data hasil pengukuran yang telah dikirim dapat diunduh dalam format *.csv* yang berguna bagi analisis dan *logger data*. Berikut merupakan langkah – langkah dalam membuat *server* pada *thingspeak*.

- Langkah pertama untuk menggunakan penyimpanan *thingspeak.com* adalah membuat akun baru dan *login* pada halaman tersebut
- Melakukan penyetelan *Channel* yang akan ditampilkan pada *server*. *Channel* berisi nilai atau data hasil pengukuran *sensor* yang telah dikirim menggunakan *Ethernet shield*. Gambar 3.24 merupakan halaman muka penyetelan *Channel*.



Gambar 3.24 Pembuatan *Channel*

- Untuk membuat *channel* baru pilih pada *New Channel*, yang akan memunculkan halaman seperti pada Gambar 3.25



ThingSpeak Channels Apps Community Support How to Buy Account Sign Out

New Channel

Name

Description

Field 1 ☒

Field 2 ☐

Field 3 ☐

Field 4 ☐

Field 5 ☐

Field 6 ☐

Field 7 ☐

Field 8 ☐

Metadata

Help

Channels store all the data that a ThingSpeak application collects. Each channel includes eight fields that can hold any type of data, plus three fields for location data and one for status data. Once you collect data in a channel, you can use ThingSpeak apps to analyze and visualize it.

Channel Settings

- Channel Name:** Enter a unique name for the ThingSpeak channel.
- Description:** Enter a description of the ThingSpeak channel.
- Field:** Check the box to enable the field, and enter a field name. Each ThingSpeak channel can have up to 8 fields.
- Metadata:** Enter information about channel data, including JSON, XML, or CSV data.
- Tag:** Enter keywords that identify the channel. Separate tags with commas.
- Latitude:** Specify the position of the sensor or thing that collects data in decimal degrees. For example, the latitude of the city of London is 51.5072.
- Longitude:** Specify the position of the sensor or thing that collects data in decimal degrees. For example, the longitude of the city of London is -0.1275.
- Elevation:** Specify the position of the sensor or thing that collects data in meters. For example, the elevation of the city of London is 35.052.
- Make Public:** If you want to make the channel publicly available, check this box.
- URL:** If you have a website that contains information about your ThingSpeak channel, specify the URL.

Gambar 3.25 Mengisi Data Channel

- d. Isi dengan lengkap data pada kolom halaman tersebut. *Server* yang dibutuhkan sebanyak 3 buah sehingga dilakukan 3x pembuatan *server*. *Channel Name* akan memuat nama unik dari *Channel Thingspeak.com* yang mengindikasikan isi data dari pembuat *Channel*. Pada *Field 1* – hingga *Field 8* merupakan data tempat yang akan diunggah *Ethernet Shield* ke *server thingspeak.com*. Pengisian kolom pada kolom halaman seperti pada Tabel 3.1.

Tabel 3.1 Konfigurasi Pengisian Channel

Form	Sensor	Mode	Konfigurasi
Nama	Monitoring Sensor Penjernihan Air	Monitoring aktuator proses penjernihan air	Monitoring mode aktuator
Field 1	Ketinggian Air Pada Tangki 1	pompa1	pompa1
Field 2	Ketinggian Air Pada Tangki 2	motor	motor
Field 3	Kekeruhan Awal	Valve1	Valve1
Field 4	Kekeruhan setelah penjernihan	Valve2	Valve2
Field 5	-	Valve3	Valve3

<i>Form</i>	<i>Sensor</i>	<i>Mode</i>	Konfigurasi
<i>Field 6</i>	-	Pompa	Pompa

Hasil tampilan dari *Field* tersebut akan berupa grafik yang menampilkan poin – poin pengukuran informasi kekeruhan, ketinggian atau kondisi aktuator. Contoh gambar tampilan dari halaman *channel sensor thingspeak.com* terdapat pada Gambar 3.26.

3.3.7 Perancangan Website

Perancangan *Website* merupakan tanggung jawab Rivaldy Hariansyah. Pada perancangan *website* dibuat beberapa halaman *website* menggunakan Bahasa pemrograman HTML dan PHP untuk menjadi sebuah *website* yang dapat diakses dan informasi mudah dibaca dan dipahami oleh pengguna. Tahapan pembuatan *website* hingga siap digunakan adalah sebagai berikut.

1. Membuat *hosting* dan *domain*

Terdapat banyak penyedia *hosting* dan *domain* berbayar atau gratis. Pada Tugas Akhir ini kami menggunakan jasa IdHostinger untuk menyewa *hosting* dan *domain*. Untuk dapat menyewa harus melakukan registrasi terlebih dahulu pada halaman <http://idhostinger.com/>. Alamat *domain/website* yang kami buat adalah <http://penjernihanair.esy.es/>

2. Membuat halaman *website*

Dalam membuat sebuah halaman *website* diperlukan aplikasi editor seperti notepad dan *word* untuk menuliskan *script* PHP dan *HTML*. Pada *website* yang kami buat terdapat beberapa halaman *website* antara lain:

- a. Halaman *Login*
- b. Halaman koneksi
- c. Halaman *header* dan *footer*
- d. Halaman grafik *sensor*
- e. Halaman Tabel *Sensor*
- f. Halaman Tabel Aktuator
- g. Halaman Tabel *Report*
- h. Halaman konfigurasi & Grafik Aktuator
- i. Halaman mengenai teori, cara kerja, *hardware* dan *software* pendukung Tugas Akhir
- j. Halaman *parsing* data *Thingspeak* ke *Android*
- k. Halaman *parsing* data *Thingspeak* ke *website*

Pada Gambar 3.27 merupakan *script* halaman tabel *sensor* yang dibuat menggunakan aplikasi Notepad+ yang terdapat pada komputer.

3. *Upload* halaman *website* ke *Hosting*

Langkah selanjutnya adalah meng-*upload* file/halaman yang telah dibuat ke *webserver/hosting*. Untuk melakukannya diperlukan sebuah aplikasi yang dapat mengakses folder ftp dari *server* yang dibuat. Pada Tugas Akhir ini digunakan aplikasi FileZilla untuk mengunggah halaman *website*. FileZilla terlebih dahulu harus melakukan koneksi dan *login* ke halaman ftp *webserver* Tampilan aplikasi FileZilla terdapat pada Gambar 3.28.

4. Membuat *database* MySQL dan mengkoneksikan dengan halaman *website*

Database diperlukan dalam sebuah *website* untuk menyimpan data. Untuk membuat *database* diperlukan *login* terlebih dahulu pada *CPanel server*. Setelah masuk *CPanel* pilih *database MySQL*. *Database* yang telah dibuat pada *phpMyAdmin* seperti pada Gambar 3.29. *Database* yang telah dibuat memiliki nama *database* dan nama *user*: u262592426_tugas dan *host MySQL* yang digunakan : mysql.idhostinger.com. dua data ini digunakan untuk dapat mengubah, menambah dan mengedit isi *database*. Apabila *database* telah dibuat maka *website* siap digunakan. Tampilan awal halaman *web* pada Gambar 3.30

Monitoring Sensor Penjernihan Air

Channel ID: 167107
Author: koyolwiny
Access: Public

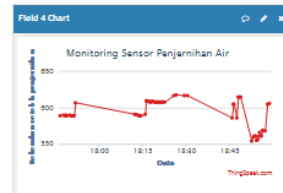
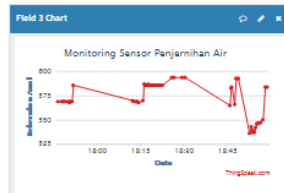
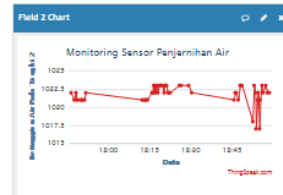
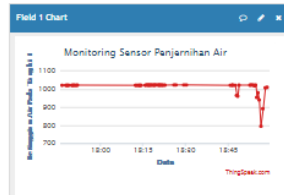
[Private View](#) [Public View](#) [Channel Settings](#) [API Keys](#) [Data Import / Export](#)

[Add Visualizations](#) [Data Export](#) [More Information](#)

[MATLAB Analysis](#) [MATLAB Visualization](#)

Channel Stats

Created: 8 months ago
Updated: 8 minutes ago
Last entry: about 24 hours ago
Entries: 555



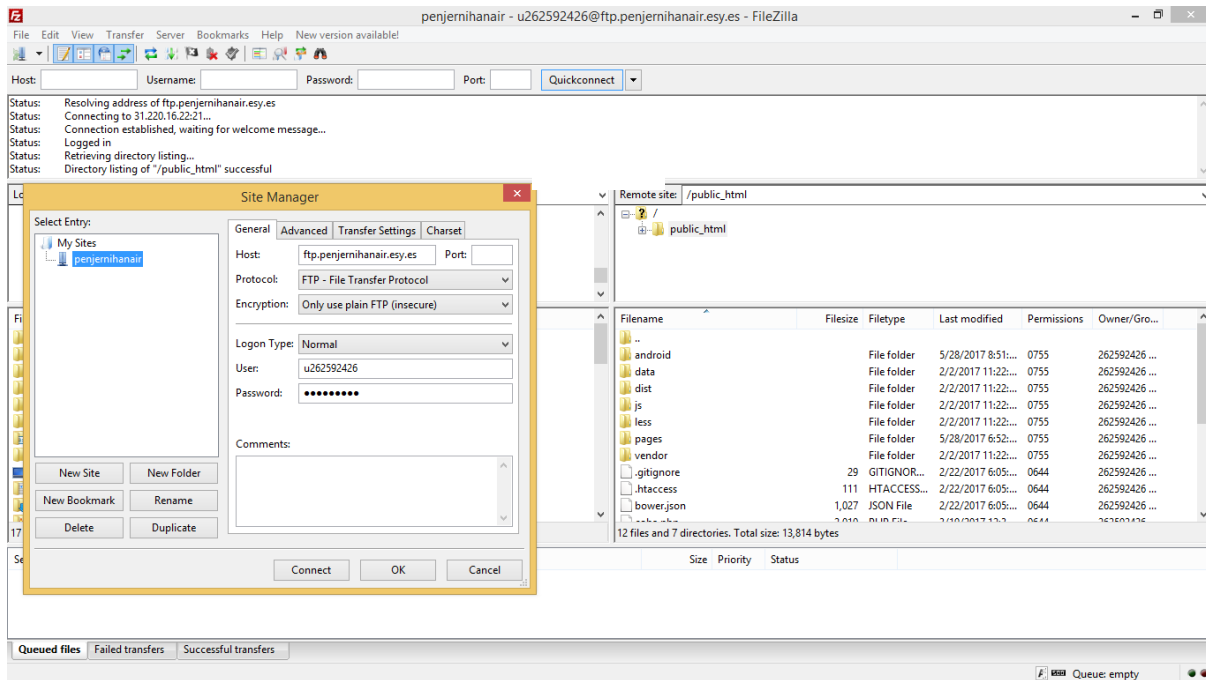
Gambar 3.26 Hasil Pembacaan *Sensor* Oleh *Thingspeak*

```

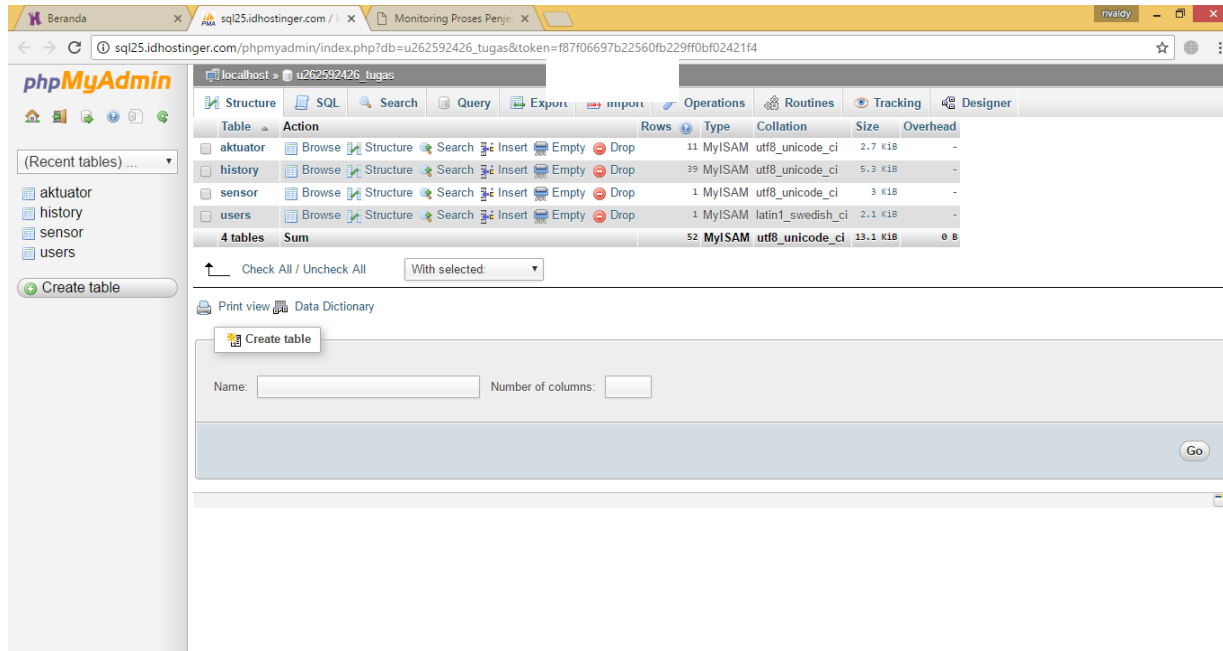
1 <?php
2     include "header.php";
3
4     <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.3.0/jquery.min.js" type="text/javascript"></script>
5     <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.3.0/jquery.min.js" type="text/javascript"></script>
6     <script type="text/javascript" src="https://ajax.googleapis.com/ajax/libs/jquery/1.3.0/jquery.min.js"></script>
7
8     <div id="page-wrapper">
9         <div class="row">
10             <div class="col-lg-12">
11                 <h1 class="page-header">Monitoring Realtime Kondisi Kekeruhan dan Level Proses Penjernihan Air</h1>
12             </div>
13             <!-- /.col-lg-12 -->
14         </div>
15         <div class="panel-body">
16             <!-- Nav tabs -->
17             <ul class="nav nav-pills">
18                 <li class="active"><a href="#home-pills" data-toggle="tab">Kekeruhan Air</a>
19                 </li>
20                 <li><a href="#profile-pills" data-toggle="tab">Level Ketinggian Tangki</a>
21                 </li>
22             </ul>
23             <!-- Tab panes -->
24             <div class="tab-content">
25                 <div class="tab-pane fade in active" id="home-pills">
26                     <h4>Pengukuran Kekeruhan Air</h4>
27                     <div class="col-lg-6">
28                         <div class="panel panel-primary">
29                             <div class="panel-heading">
30                                 Tingkat Kekeruhan Awal Proses
31                             </div>
32                             <!-- /.panel-heading -->
33                             <div class="panel-body">
34                                 <iframe width="450" height="260" style="border: 1px solid #cccccc;" src="https://thingspeak.com/channels/167107/charts/1?bgcolor=%23ffff"
35                                 </div>
36                             <!-- /.panel-body -->
37                         </div>

```

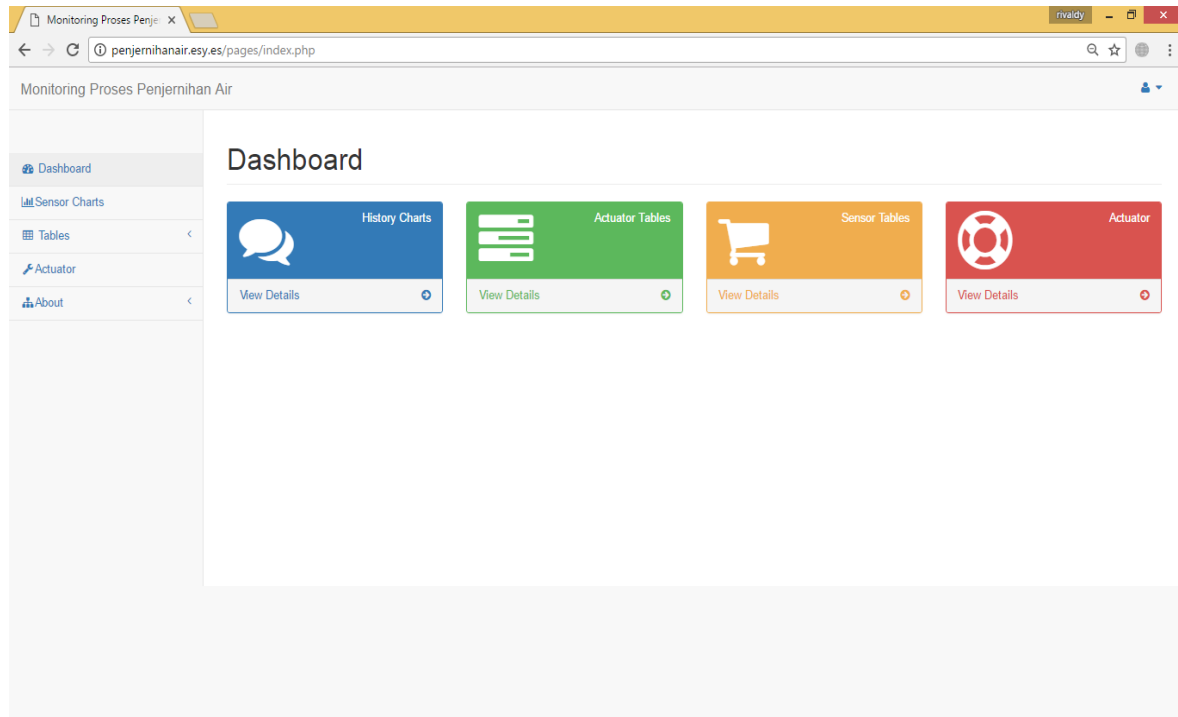
Gambar 3.27 Script Pada Notepad+



Gambar 3.28 Aplikasi FileZilla Untuk Mengirim Data ke Server



Gambar 3.29 *Phpmyadmin Untuk Membuat Database MySQL*

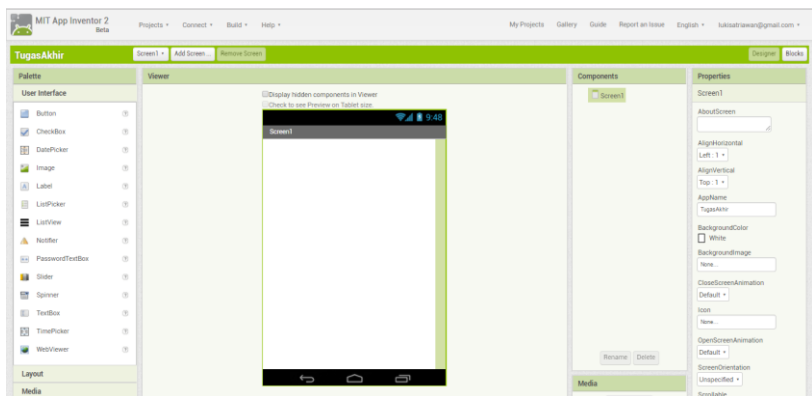


Gambar 3.30 Halaman *Website* Setelah Diupload

3.3.8 Perancangan Aplikasi *Android*

Perancangan aplikasi *Android* merupakan tanggung jawab Achmad Luki Satriawan. Pada perancangan aplikasi *Android* dilakukan dengan menggunakan *App Inventor*. Sebelum dapat menggunakan aplikasi *Android*, langkah pertama yaitu dengan melakukan desain untuk tampilan pada *Android* yang akan digunakan untuk proses *monitoring*. Kemudian untuk dapat menjalankan aplikasi pengguna perlu mengatur pada diagram blok Berikut adalah tahapan untuk membuat aplikasi *Android* :

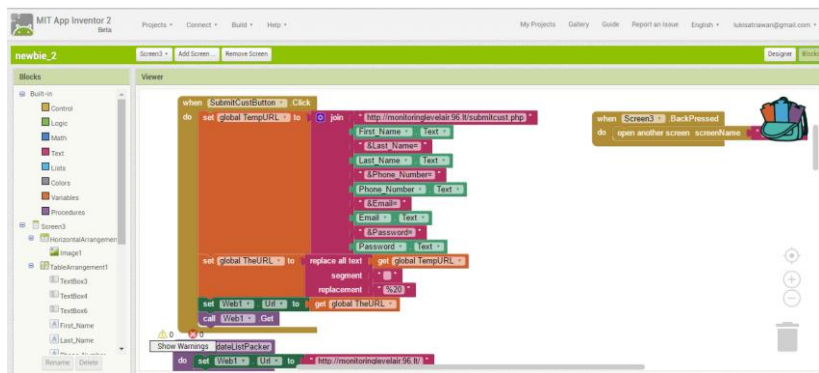
1. Pengguna harus memiliki *account google*. Kemudian mengakses alamat <http://ai2.appinventor.mit.edu>
2. Setelah akses alamat *website* dijalankan maka akan muncul tampilan antar muka dari *App Inventor*. Pengguna dapat mengatur tampilan desain dari aplikasi *Android* yang akan dibuat. Antar Muka *Android* pada *App Inventor* terdapat pada Gambar 3.31



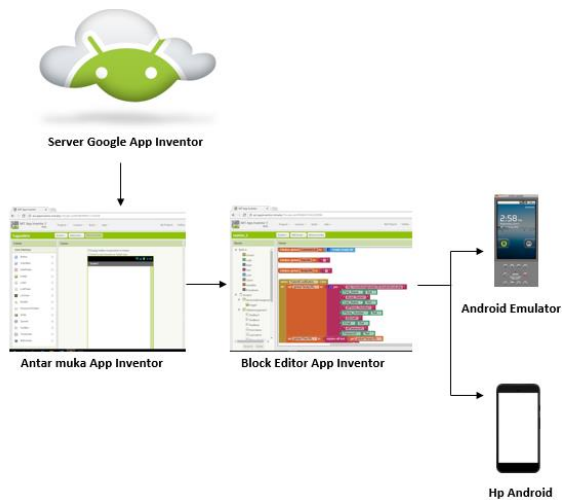
Gambar 3.31 Antar Muka *Android*

3. Tahap selanjutnya memprogram pada *block editor*. *Block editor* berguna untuk memprogram *visual block (puzzle)* pada komponen yang telah dipilih oleh pengguna pada tampilan antar muka desainer dengan cara *drag and drop*. Gambar *Block Editor* terdapat pada Gambar 3.32
4. Kemudian mensimulasikan aplikasi *Android* yang telah dibuat dengan mentransfer aplikasi *android* pada *handphone* pengguna. Akan tetapi, *App Inventor* juga menyediakan *Android emulator* apabila pengguna tidak memiliki *handphone Android*.

Secara umum skema alur dari pembuatan aplikasi *Android* yang dibuat adalah seperti pada Gambar 3.33



Gambar 3.32 Block Editor Android

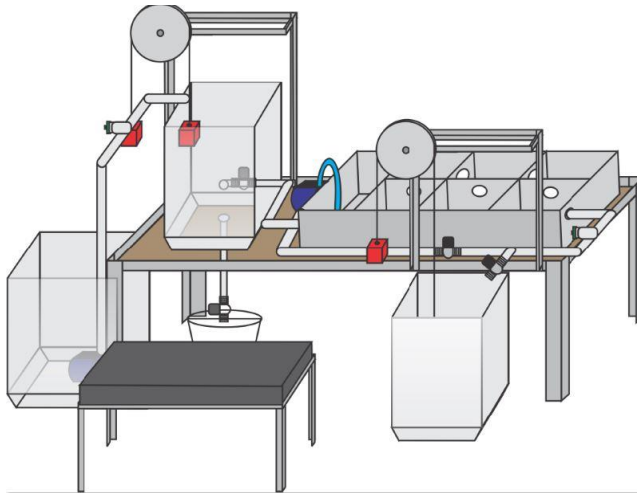


Gambar 3.33 Skema Alur Pembuatan *Android*

3.3.9 Perancangan Sistem Telemetri *Level* Dan Kekeruhan Air Pada Alat Penjernihan Air Sungai

Rancangan keseluruhan alat untuk Telemetri *level* dan kekeruhan air pada proses penjernihan air sungai ditunjukkan pada Gambar 3.34. Pada alat tersebut terdapat 2 buah *sensor level* yaitu pada tangki penampungan air sungai dan tangki penampungan hasil proses penjernihan air.

Sebelum memasuki tangki penampungan dilakukan pembacaan tingkat kekeruhan air menggunakan *turbidity sensor* untuk mengetahui tingkat kekeruhan air sungai yang digunakan. Pada akhir proses penjernihan dilakukan pembacaan tingkat kekeruhan air kembali, apabila tingkat kekeruhan air masih dibawah nilai 100 NTU maka air dikembalikan kembali ke awal proses filtrasi. Air bersih yang telah dihasilkan pada proses filtrasi akan ditampung pada tangki penampungan akhir untuk diendapkan. Air tersebut dapat digunakan untuk proses aktivitas sehari – hari seperti mandi dan mencuci.



Gambar 3.34 Rancangan Alat Keseluruhan

Hasil pembacaan tegangan 2 buah *sensor level* dan 2 buah *sensor* kekeruhan akan diolah oleh Arduino dan dikonversi kesatuan *centimeter* untuk *sensor* ketinggian dan NTU untuk *sensor* kekeruhan. *Ethernet Shield* pada Arduino berperan dalam mengirim nilai *sensor* yang telah dikonversi ke *webserver* secara terus menerus. Operator / pengguna dapat memantau tingkat ketinggian tangki penampungan dan kekeruhan air melalui aplikasi *monitoring* penjernihan air melalui *Android*, aplikasi HMI LabVIEW melalui Windows atau melalui *browser* pada alamat <http://penjernihanair.esy.es/>. Selain itu kondisi dari aktuatur pada alat penjernihan air seperti pompa air, *valve* dan motor DC dapat termonitor dan ter-record setiap perubahannya. Operator dapat mengaktifkan atau menonaktifkan setiap aktuatur secara jarak jauh tanpa harus berada didekat alat.

BAB IV

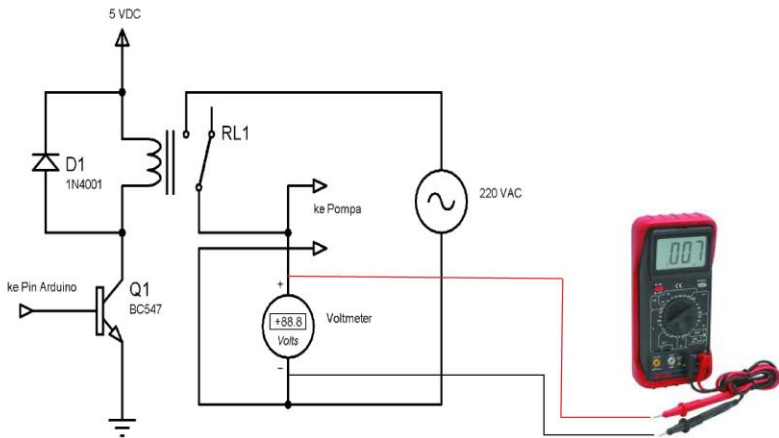
HASIL SIMULASI DAN UJI COBA

Pada bab ini akan dilakukan pembahasan untuk pengujian pada tiap-tiap rangkaian yang menyusun alat ini. Pada pengujian yang dilakukan kali ini dimaksudkan untuk memastikan kinerja pada masing-masing alat yang telah terpasang dengan harapan pengujian yang dilakukan dapat berfungsi sesuai dengan yang diharapkan. Untuk mengetahui kekurangan dari alat dilakukan pengujian pada masing-masing perangkat yaitu perangkat keras (*hardware*) dan perangkat lunak (*software*). Pengujian yang dilakukan meliputi pengujian pompa air, *power supply*, pengujian *sensor level*, pengujian *relay*, pengujian Motor DC, pengujian *sensor turbidity*, pengujian *ethernet shield*, pengujian *Solenoid Valve*, pengujian HMI LabVIEW, pengujian *Webserver*, pengujian aplikasi *Android*, pengujian proses penjernihan air dan pengujian alat secara keseluruhan.

4.1 Pengujian Pompa Air

Pengujian pompa air merupakan tanggung jawab Achmad Luki Satriawan. Pompa diperlukan untuk mensuplai air ke dalam tangki pengadukan maupun ke akuarium filter. Untuk itu diperlukan pengujian agar sistem berjalan lancar. Rangkaian untuk pengujian pompa dapat dilihat pada Gambar 4.1. Metode pengujian pompa air yaitu dengan menghubungkan *power supply* 5 Volt ke arduino yang terhubung dengan *Relay 4 Channel*. Pin *VCC relay* dihubungkan dengan pin *VCC Arduino* dan Pin data 1 dihubungkan dengan pin *Digital Output*. Terminal *Relay* dihubungkan dengan tegangan 220V dan keluaran *NO Relay* dihubungkan salah satu kabel pompa air. Kabel lain dari pompa air dihubungkan dengan sisi netral pada *power supply* yang terhubung dengan listrik PLN. Pompa air diletakan di dalam bak air yang telah terisi air penuh. Pengujian dilakukan dengan memberikan data *digital High* dan *Low* pada *relay*. Program arduino untuk pompa terdapat pada lampiran A.2. Program Arduino untuk Pengujian Pompa Air.

Apabila pompa aktif, maka air akan terdorong keluar dari bak air dan menuju ke tempat dimana air akan dialirkan. Hasil pengujian dapat dilihat pada Tabel 4.1



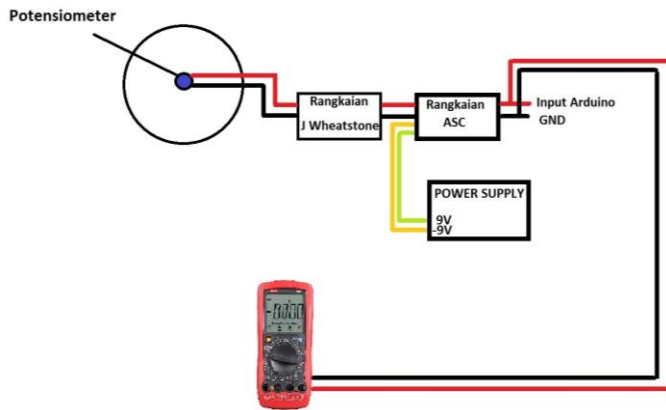
Gambar 4.1 Rangkaian Pengujian Pompa

Tabel 4.1 Data Pengujian Pompa Air

No	Pin Input (Volt)	Tegangan Keluaran pin NO Relay (Volt)	Keadaan Pompa Air
1	0 Volt	0 Volt	Mati
2	5 Volt	220 Volt	Aktif

4.2 Pengujian *Sensor Ketinggian*

Pengujian *Sensor Ketinggian* merupakan tanggung jawab Achmad Luki Satriawan. Pengujian *sensor* ketinggian bertujuan untuk mengetahui tegangan *output* yang dihasilkan pada saat air melakukan pengisian pada tangki. Bentuk fisik *sensor* ketinggian air pada alat seperti pada Gambar 4.3. Pengujian dilakukan dengan menghubungkan catu daya +9 Volt dan -9 Volt ke rangkaian ASC yang terhubung dengan *sensor* ketinggian kemudian air sungai di pompa ke tangki pengaduk, disaat air mengisi tangki menyebabkan bandul naik turun tergantung *level* didalam air sehingga mempengaruhi nilai resistansi pada potensiometer yang terpasang pada alat seperti pada Gambar 4.2. Pergerakan bandul akan mempengaruhi nilai resistansi yang kemudian dikonversi ke tegangan melalui rangkaian jembatan *Wheatstone*. Kemudian *output* rangkaian jembatan *Wheatstone* masuk ke rangkaian *signal conditioning* dan ke *input* kontroler yaitu arduino.



Gambar 4.2 Rangkaian Pengujian *Sensor* Ketinggian



Gambar 4.3 Bentuk *Hardware* *Sensor* Ketinggian

Tabel 4.2 Data Pengujian Tangki Penampungan Awal

No	Ketinggian Tangki (Cm)	Volume Terukur (Cm³)	Persentase ketinggian air (%)	Vout ASC (Volt)
1	0	0	0	0
2	1	0,21	4	0,56
3	2	0,45	8	0,79
4	3	0,86	12	1,02
5	4	1,40	16	1,35
6	5	1,85	20	1,65
7	6	2,3	24	1,81
8	7	2,75	28	2,03
9	8	3,26	32	2,26
10	9	3,65	36	2,40
11	10	4,1	40	2,62
12	11	4,56	44	2,74
13	12	5,01	48	2,88
14	13	5,46	52	3,11
15	14	5,91	56	3,20
16	15	6,36	60	3,36
17	16	6,81	64	3,53
18	17	7,26	68	3,67
19	18	7,72	72	3,79
20	19	8,17	76	3,90
21	20	8,62	80	4,06
22	21vou	9,07	84	4,16

No	Ketinggian Tangki (Cm)	Volume Terukur (Cm ³)	Persentase ketinggian air (%)	Vout ASC (Volt)
23	22	9,52	88	4,25
24	23	9,97	92	4,41
25	24	10,43	96	4,45
26	25	10,88	100	4,56
Rata-rata				2,80

Berdasarkan Tabel 4.2 rata-rata penurunan tegangan setiap 1 Cm bernilai 2,80 Volt. Pada perhitungan *volume* terukur, terdapat pada Persamaan 4.4 dan Persamaan 4.5. Setelah di dapatkan data pengukuran, dengan menggunakan metode regresi linier *slope intercept*, maka diperoleh fungsi seperti berikut :

$$y = 5,777304121x - 3,656453177 \dots \dots \dots (4.1)$$

Keterangan :

y = ketinggian air (Cm)

x = Vout (Volt)

Perhitungan nilai *error* :

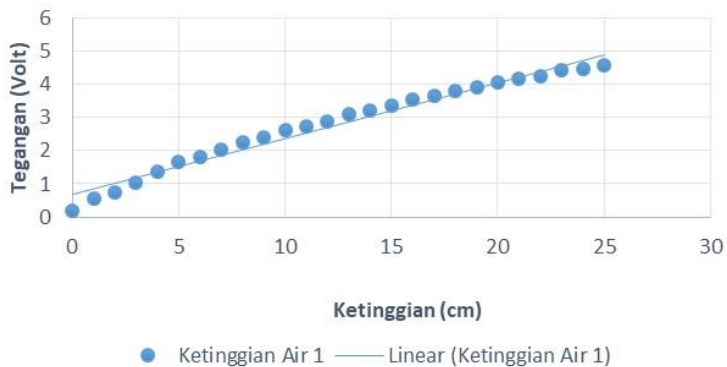
$$\text{Nilai Error : } \left(\frac{\text{Nilai Sebenarnya} - \text{Nilai Terukur}}{\text{Nilai Sebenarnya}} \right) \times 100\%$$

Tabel 4.3 Data Ketinggian Tangki Penampungan Awal

No	Ketinggian Tangki (Cm)	Ketinggian Air Yang Terbaca Sensor	Error (%)
1	0	0,13	3,26
2	1	0,34	142,1
3	2	0,67	66,17
4	3	2,23	25,45
5	4	4,14	3,57
6	5	5,87	17,52
7	6	6,80	13,34
8	7	8,07	15,30
9	8	9,40	17,50
10	9	10,20	13,43
11	10	11,48	14,80
12	11	12,11	10,66

No	Ketinggian Tangki (Cm)	Ketinggian Air Yang Terbaca <i>Sensor</i>	<i>Error (%)</i>
13	12	12,47	8,18
14	13	14,31	10,08
15	14	14,83	5,93
16	15	15,75	5,03
17	16	16,73	4,60
18	17	17,54	3,21
19	18	18,23	1,33
20	19	18,87	0,65
21	20	19,79	1,00
22	21	20,37	2,96
23	22	20,89	5,01
24	23	21,82	5,12
25	24	22,06	8,11
26	25	22,68	9,24

Pada Tabel 4.3 ketinggian 0 sampai dengan 3 Cm persentase *error* bernilai sangat tinggi. Namun pada ketinggian 4 sampai dengan 11 Cm %*error* masih terlihat lumayan besar yaitu sekitar 3-10%. Dan %*error* pada ketinggian 12 sampai dengan 25 Cm memiliki nilai %*error* hampir mendekati nilai kecil yaitu sekitar 0,65-5,2%. Pada ketinggian alat ini menggunakan ketinggian 25 Cm dikarenakan menyesuaikan dengan tinggi tangki yang digunakan. Grafik Linierisasi pada Persamaan 4.1 terdapat pada Gambar 4.4



Gambar 4.4 Pengujian *Sensor* Ketinggian Pada Tangki Penampungan Awal

Tabel 4.4 Data Hasil Pengujian Tangki Akhir

No	Ketinggian Tangki (Cm)	Volume Terukur (Cm ³)	Persentase ketinggian air (%)	Vout ASC (Volt)
1	0	0	0	0
2	1	0,45	4	0,27
3	2	0,90	8	0,55
4	3	1,35	12	0,71
5	4	1,80	16	1,08
6	5	2,25	20	1,41
7	6	2,70	24	1,60
8	7	3,16	28	1,82
9	8	3,61	32	1,99
10	9	4,06	36	2,13
11	10	4,51	40	2,49
12	11	4,96	44	2,73
13	12	5,41	48	2,92
14	13	5,86	52	3,10
15	14	6,32	56	3,20
16	15	6,77	60	3,37
17	16	7,22	64	3,61
18	17	7,67	68	3,76
19	18	8,12	72	3,91
20	19	8,57	76	4,01
21	20	9,03	80	4,11
22	21	9,48	84	4,27
23	22	9,93	88	4,45
24	23	10,38	92	4,54

No	Ketinggian Tangki (Cm)	Volume Terukur (Cm ³)	Persentase ketinggian air (%)	Vout ASC (Volt)
25	24	10,83	96	4,61
26	25	11,28	100	4,71
Rata-rata				2,74

Berdasarkan Tabel 4.4 rata-rata penurunan tegangan setiap 1 Cm bernilai 2,74 Volt. Pada perhitungan *volume* terukur, terdapat pada Persamaan 4.4 dan Persamaan 4.5. Setelah di dapatkan data pengukuran, dengan menggunakan metode regresi linier *slope intercept*, maka diperoleh fungsi seperti berikut :

$$y = 5,20109989x - 1,797023429 \dots \dots \dots (4.2)$$

Keterangan :

y = ketinggian air (Cm)

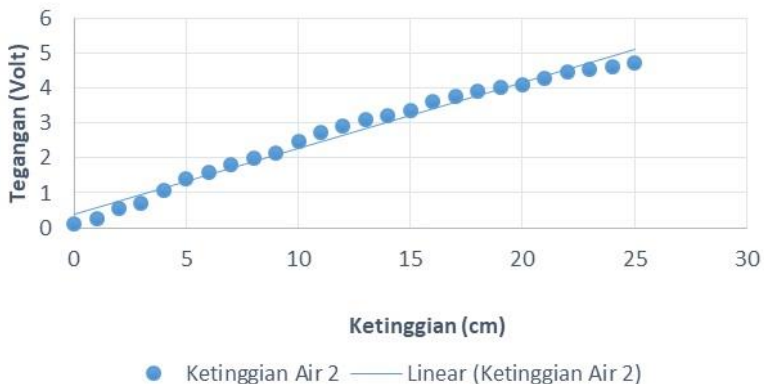
x = Vout (Volt)

Tabel 4.5 Data Perbandingan Ketinggian Tangki Penampungan Akhir

No	Ketinggian Tangki (Cm)	Ketinggian Air Yang Terbaca <i>Sensor</i>	Error (%)
1	0	0,12	3,16
2	1	0,98	60,72
3	2	1,06	46,82
4	3	1,89	36,80
5	4	3,82	4,49
6	5	5,53	10,73
7	6	6,52	8,74
8	7	7,66	9,55
9	8	8,55	6,91
10	9	9,28	3,12
11	10	11,15	11,53
12	11	12,40	12,74
13	12	13,39	11,58
14	13	14,32	10,20
15	14	14,84	6,04
16	15	15,37	4,87
17	16	16,97	6,11
18	17	17,75	4,46
19	18	18,53	2,99
20	19	19,05	0,31
21	20	19,57	2,10

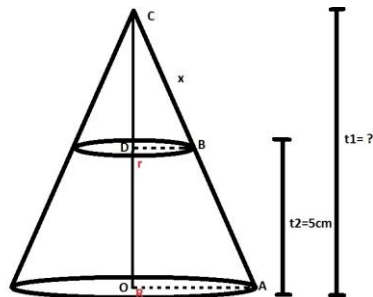
No	Ketinggian Tangki (Cm)	Ketinggian Air Yang Terbaca <i>Sensor</i>	<i>Error (%)</i>
22	21	20,41	2,80
23	22	21,34	2,96
24	23	21,81	5,14
25	24	22,18	7,58
26	25	22,80	8,78

Pada Tabel 4.5, pada ketinggian 0 sampai dengan 3 Cm, %error bernilai sangat tinggi. Namun pada ketinggian 10 sampai dengan 11 Cm %error masih terlihat lumayan besar yaitu sekitar 10-12%. Dan %error pada ketinggian 14 sampai dengan 25 Cm memiliki nilai %error hampir mendekati nilai kecil yaitu sekitar 0,31-8,78%. Pada ketinggian alat ini menggunakan ketinggian 25 Cm dikarenakan menyesuaikan dengan tinggi tangki yang digunakan. Grafik Linierisasi pada Persamaan 4.2 terdapat pada Gambar 4.5



Gambar 4.5 Pengujian *Sensor* Ketinggian Pada Tangki Penampungan Akhir

4.2.1 Perhitungan Volume Tangki



Gambar 4.6 Limas Terpancung

Untuk dapat mengetahui *volume* pada limas terpancung yaitu pada Gambar 4.6, tahap awal yaitu menentukan nilai X yaitu merupakan sisi miring yang belum diketahui. Berikut merupakan perhitungan yang untuk mengetahui nilai x

$$\triangle_{DBC} = \triangle_{OAC}$$

$$\text{Sehingga : } \frac{DB}{OA} = \frac{CD}{CO} = \frac{CB}{CA}$$

Diketahui BD=7,5 Cm ; OA= 10,5 Cm

Misal CB=x , Maka

$$\frac{x}{x + 5} = \frac{7,25}{10,5}$$

$$10,5x = 7,25x + 36,25$$

$$10,5x - 7,25x = 36,25$$

$$3,25x = 36,25$$

$$x = 11,15 \dots \dots \dots (4.3)$$

Setelah Persamaan 4.3 didapat maka tinggi total dari nilai x telah diketahui, kemudian dilakukan perhitungan *volume* yang berbentuk limas dengan alas yang berbentuk persegi sehingga perhitungan luas alas yang sebelumnya berbentuk lingkaran diganti dengan persegi agar sesuai dengan kondisi dilapangan tangki yang digunakan (Seperti pada Gambar 3.11)

$$\begin{aligned}
Volume_{Tot} &= V_{Besar} - V_{kecil} \\
&= \frac{1}{3} S^2 t_1 - \frac{1}{3} S^2 t_2 \\
&= \frac{1}{3} 22.21,5.11,15 - \frac{1}{3} .14,5.14.5 \\
&= 1813,16 - 338,33 \\
&= 1474,83 \text{ Cm}^3 \\
&= 1,47483 \text{ Liter}.....(4.4)
\end{aligned}$$

Selanjutnya dilakukan perhitungan *volume* tangki berbentuk balok. Pada perhitungan berikut untuk mengetahui *volume* yang tertampung pada tangki pada tangki yang berbentuk balok yaitu diketahui P=25Cm, L=21Cm, T=21,5 Cm (Seperti pada Gambar 3.11) sehingga

$$\begin{aligned}
V_{balok} &= P \times L \times T \\
&= 25 \times 21 \times 21,5 \\
&= 11287,5 \text{ Cm}^3 \\
&= 11,2875 \text{ Liter}.....(4.5)
\end{aligned}$$

Sehingga setelah Persamaan 4.4 dan Persamaan 4.5 diketahui ,*volume* total dari tangki adalah

$$\begin{aligned}
V_{total} &= V_{limas} + V_{balok} \\
&= 1,47 + 11,28 \\
&= 12,75 \text{ Liter}
\end{aligned}$$

4.2.2 Rangkaian Jembatan *Wheatstone*

Pada rangkaian jembatan *Wheatstone* ini berguna untuk mengetahui tegangan terukur dari potensiometer yang terpasang pada alat. Dengan naik turun dari bandul yang terpasang pada alat menyebabkan tegangan terukur mengalami perubahan. Gambar rangkaian Jembatan *Wheatstone* terdapat pada Gambar 4.7.

Tabel 4.6 Pengukuran Tegangan *output J.Wheatstone* Tangki Penampungan Awal

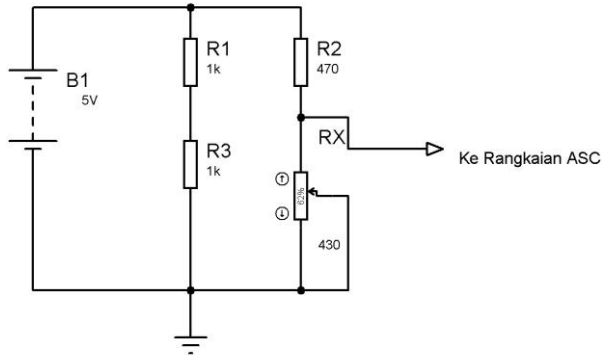
No	Keterangan	<i>Vout J.Wheatstone</i>
1	Tegangan Minimal	0
2	Tegangan Maksimal	2,11

Pada Tabel 4.6 merupakan pengukuran yang dilakukan pada tangki penampungan awal yaitu tegangan terukur saat putaran *sensor* berada pada ketinggian maksimal.

Tabel 4.7 Data Pengukuran Tegangan *output J Wheatstone* Tangki Akhir

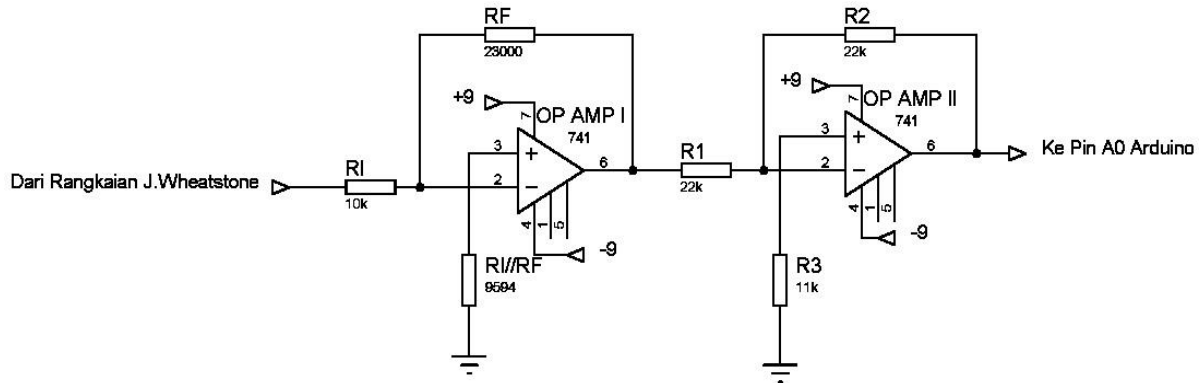
No	Keterangan	<i>Vout J.Wheatstone</i>
1	Tegangan Minimal	0
2	Tegangan Maksimal	2,08

Pada Tabel 4.7 merupakan pengukuran tegangan yang dilakukan pada tangki penampungan akhir yaitu pada keadaan terendah dari tangki saat terdapat air hingga keadaan tertinggi dari tangki yang diukur.



Gambar 4.7 Rangkaian Jembatan *Wheatstone*

4.2.3 Rangkaian *Signal Conditioning*



Gambar 4.8 Rangkaian *Signal Conditioning*

Rangkaian *signal conditioning* yang terdapat pada Gambar 4.8 berguna untuk menguatkan nilai tegangan yang terukur dari rangkaian jembatan *Wheatstone* yang memiliki nilai tidak sesuai atau nilai yang belum memenuhi syarat untuk dapat di masukkan pada *input* kontroler sehingga perlu dilakukan penguatan untuk dapat terbaca tegangan dari *sensor* ke *input* kontroler. Rangkaian pada Gambar 4.8 berguna untuk mengubah tegangan dari tegangan terukur (0-2,11 Volt) menjadi (0-5Volt) Berikut merupakan perhitungan rangkaian *signal conditioning* :

$$\frac{Y - Y1}{Y2 - Y1} = \frac{X - X1}{X2 - X1}$$

$$\frac{Y - 0}{5 - 0} = \frac{X - 0}{2,11 - 0}$$

$$Y = \frac{5}{2,11} X$$

$$\frac{R_f}{R_i} = m \quad ; R_i = 10k \text{ Ohm}$$

$$R_f = 10k \cdot 2,369$$

$$R_f = 23690 \Omega$$

$$\frac{1}{R_p} = \frac{1}{R_f} + \frac{1}{R_i}$$

$$\frac{1}{R_p} = \frac{1}{23690} + \frac{1}{10k}$$

$$\frac{1}{R_p} = \frac{246900}{236900000}$$

$$R_p = 9594 \Omega \dots \dots \dots (4.6)$$

Dari perhitungan yang dilakukan seperti pada Persamaan 4.5 dilakukan pengukuran pada rangkaian *signal conditioning* (Seperti pada Gambar 4.2) didapat pengukuran yaitu dengan tegangan dengan rentang 0,38-4,56 Volt.

Tabel 4.8 Data Pengukuran Rangkaian *Signal Conditioning* Tangki Awal

No	Keterangan	Vout ASC
1	Tegangan Minimal	0,38
2	Tegangan Maksimal	4,56

Pada Tabel 4.8 merupakan hasil pengukuran dari rangkaian *Signal Conditioning* yang terukur pada tangki penampungan awal yaitu didapat

tegangan minimal 0,38 Volt dan tegangan tertinggi dari *plant* yang terukur yaitu 4,56 Volt.

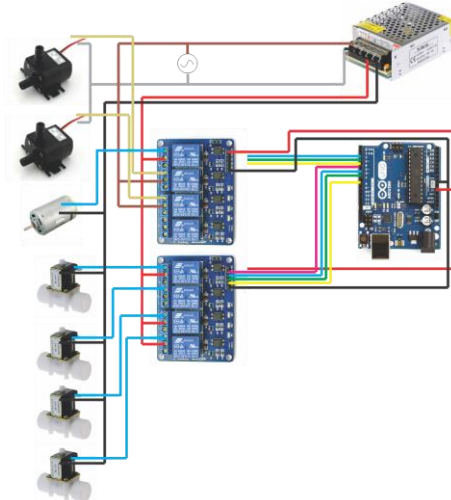
Tabel 4.9 Data Pengukuran *Rangkaian Signal Conditioning* Tangki Akhir

No	Keterangan	<i>Vout</i> ASC
1	Tegangan Minimal	0,10
2	Tegangan Maksimal	4,73

Dari perhitungan yang dilakukan seperti pada Persamaan 4.5 dilakukan pengukuran pada rangkaian *signal conditioning* (Seperti pada Gambar 4.2) didapat pengukuran yang terdapat pada Tabel 4.9 dengan tegangan dengan rentang 0,10-4,73 Volt. Untuk pengukuran detail dapat dilihat pada Tabel 4.2 untuk tangki penampungan awal dan Tabel 4.4 untuk tangki penampungan akhir.

4.3 Pengujian Relay

Pengujian *Relay* merupakan tanggung jawab Rivaldy Hariansyah. Pada Pengujian ini akan dilakukan pengujian *driver relay* untuk mengetahui aktif atau nonaktif. Metode pengukuran dilakukan dengan cara menghubungkan *VCC* pada *relay* dengan *VCC* arduino, *GND* pada *relay* dengan *GND* pada arduino, dan memberi tegangan *input* dari *Power Supply* yang diatur dari 0-5 Volt yang diberikan pada rangkaian *relay* dan dilakukan pengukuran terhadap masing-masing pin untuk mengetahui tegangan kerja yang diberikan pada *relay* dengan melihat indikator aktif atau padam. *Program* pengujian *Relay* terdapat pada lampiran A2. Gambar rangkaian terdapat pada Gambar 4.9.



Gambar 4.9 Pengujian *Relay*

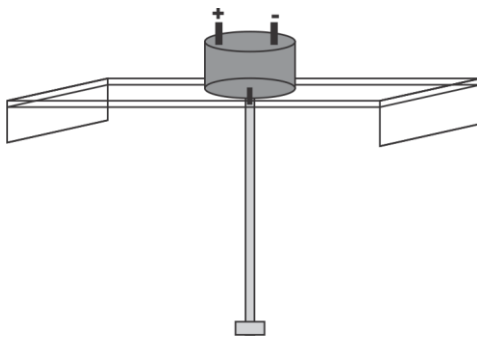
Tabel 4.10 Data Hasil Pengujian *Relay*

No	Tegangan <i>input</i> (Volt)	Keterangan	
		Kondisi Relay	Indikator Relay
1	1,25	Mati	Padam
2	1,5	Mati	Padam
3	1,75	Mati	Padam
4	2	Mati	Padam
5	2,25	Mati	Menyala Redup
6	2,5	Mati	Menyala Redup
7	3	Mati	Menyala Terang
8	3,25	Mati	Menyala Terang
9	3,3	Hidup	Menyala Terang
10	3,5	Hidup	Menyala Terang
11	3,75	Hidup	Menyala Terang
12	4	Hidup	Menyala Terang

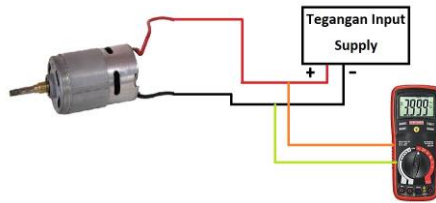
Pada Tabel 4.10 merupakan hasil pengujian *relay* yang diberikan tegangan *input* untuk mengetahui tegangan kerja dari *relay* yang digunakan pada pembuatan *prototype* Tugas Akhir.

4.4 Pengujian Motor DC

Pengujian Motor DC merupakan tanggung jawab Achmad Luki Satriawan. Pada pengujian Motor DC bertujuan untuk mengetahui tegangan kerja dari motor ketika dinyalakan. Pengujian dilakukan untuk mengetahui apakah hasil pengujian sesuai dengan yang diharapkan. Pada pengujian diberikan tegangan *input* dengan *range* 0-12 Volt dari *power supply*. Langkah yang dilakukan yaitu memasang *Voltmeter* sisi positif dan sisi negatif pada tegangan keluaran *power supply* secara paralel. Motor DC dipasang pada batang akrilik untuk menyangga agar motor DC dapat berdiri di sisi atas tangki air seperti pada Gambar 4.10. Pada bagian poros penggerak dipasang sebuah batang akrilik panjang yang berperan sebagai pengaduk air. Pengujian dilakukan dalam kondisi motor dengan beban dan motor tanpa beban. Cara pengujian motor DC pada Gambar 4.11 Pengujian dilakukan dengan kondisi tangki kosong dan tangki air diisi penuh dengan air. Dari pengujian Motor DC didapatkan hasil pada Tabel 4.11 dan Tabel 4.12 yaitu saat pengujian diberi beban dengan pengujian tanpa beban untuk mengetahui tegangan kerja dari motor DC.



Gambar 4.10 Kontruksi Motor DC pada Tangki Air



Gambar 4.11 Pengujian Motor DC

Tabel 4.11 Data Hasil Pengujian Motor DC Tanpa Beban

No	Tegangan <i>Input</i>	Keterangan
1	0	Tidak Aktif
2	1	Aktif
3	3	Aktif
4	4	Aktif
5	7	Aktif
6	12	Aktif

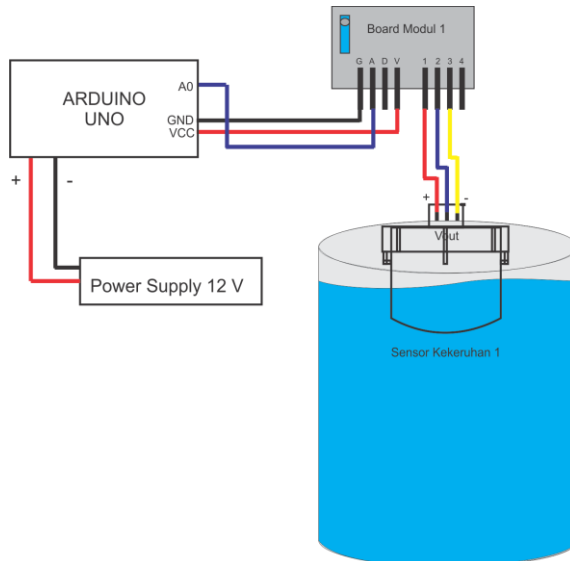
Tabel 4.12 Data Hasil Pengujian Motor DC Dengan Beban

No	Tegangan <i>Input</i>	Keterangan
1	0	Tidak Aktif
2	1	Tidak Aktif
3	3	Tidak Aktif
4	4	Aktif
5	7	Aktif
6	12	Aktif

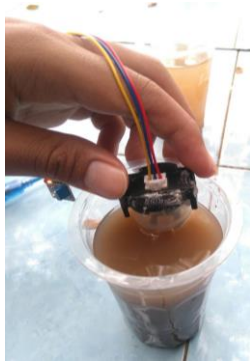
4.5 Pengujian *Sensor Kekeruhan*

Pengujian *sensor kekeruhan* merupakan tanggung jawab Rivaldy Hariansyah. Pada pengujian *sensor kekeruhan* ini memiliki tujuan untuk pembacaan satuan NTU pada *sensor*. *Sensor kekeruhan* yang digunakan adalah *turbidity sensor* DF-Robot. *Flowchart program* pengujian *sensor kekeruhan* terdapat pada Gambar 3.15. Kerja dari *sensor kekeruhan* ini yaitu *transmitter* mengirimkan cahaya ke *receiver*, tingkat intensitas cahaya akan memengaruhi tegangan yang dikeluarkan *sensor*. Pada *sen-*

sor ini LED berperan sebagai *transmitter* dan *photodiode* sebagai *receiver*. Semakin tinggi padatan yang ada didalam air maka semakin tinggi pula tingkat kekeruhan air. Satuan tingkat kekeruhan air yang digunakan adalah NTU (*Nephelometric Turbidity Units*). Metode pada alat ukur kekeruhan didasarkan pada perbandingan intensitas cahaya yang tersebar pada *sample*. Semakin tinggi intensitas cahaya tersebar maka semakin tinggi kekeruhan. Pada alat ukur Oakton T100 *Turbidity Meter* yang terdapat pada Gambar 4.16 digunakan formazin polimer digunakan sebagai referensi utama *suspense* standar dalam pengujian kalibrasi alat ukur. Kekeruhan yang digunakan untuk kalibrasi sebanyak 4 larutan standar yaitu dengan kekeruhan 800 NTU, 100 NTU, 20 NTU dan 0,02 NTU. *Sample* Air pada Gambar 4.15. kemudian diukur menggunakan *turbidity meter* lalu diukur *sensor* kekeruhan. Pada Gambar 4.12 dan Gambar 4.13 merupakan cara menggunakan *sensor* untuk mendapatkan hasil pengukuran. Pada Gambar 4.14 merupakan letak *sensor* pada *prototype*. Hasil pengukuran kedua alat ini digunakan sebagai *standart* pembandingan alat ukur *sensor* dengan alat ukur acuan. Data pengukuran *sensor* dan alat acuan terdapat pada Tabel 4.13.



Gambar 4.12 Rangkaian Pengujian *Sensor Kekeruhan*



Gambar 4.13 Pengujian Kekeruhan Air Menggunakan *Sensor Kekeruhan*



Gambar 4.14 Hardware *Sensor Kekeruhan* Pada *Prototype*



Gambar 4.15 Air Untuk Pengujian Tingkat Kekeruhan



Gambar 4.16 Alat Ukur *Turbidity* Meter merk Oakton T100

Pada Gambar 4.15 merupakan air yang digunakan untuk pengujian tingkat kekeruhan. Pada nomor 1 merupakan air sungai keruh, nomor 2 merupakan air sungai semi keruh, nomor 3 merupakan air sungai jernih, nomor 4 merupakan air Sumur dan nomor 5 merupakan air PDAM. Cara melakukan pengujian alat *turbidity* meter yaitu lakukan kalibrasi alat terlebih dahulu dengan 6 larutan *standart*. Apabila alat telah terkalibrasi dengan tepat maka selanjutnya tempatkan air *sample* pada gelas yang telah disediakan alat. Klik tombol *Read* dan tunggu proses pembacaan selama beberapa detik. Hasil pembacaan akan ditampilkan pada LCD. Lakukan proses tersebut pada semua *sample* air yang telah disiapkan. Untuk melakukan pengujian *sensor* kekeruhan yang harus dilakukan adalah melakukan *wiring sensor* seperti pada Gambar 4.12. Unduh *program AnalogRead* pada Arduino. Masukkan sebagian dari *sensor* ke dalam air seperti pada Gambar 4.13. Hasil pembacaan kekeruhan akan diterima pada pin *Analog* Arduino. Ukur pin *Analog* menggunakan *Voltmeter*. Pada Tabel 4.13 merupakan tabel hasil pengujian

menggunakan *Voltmeter* dan parameter pembanding merupakan *turbidity* meter.

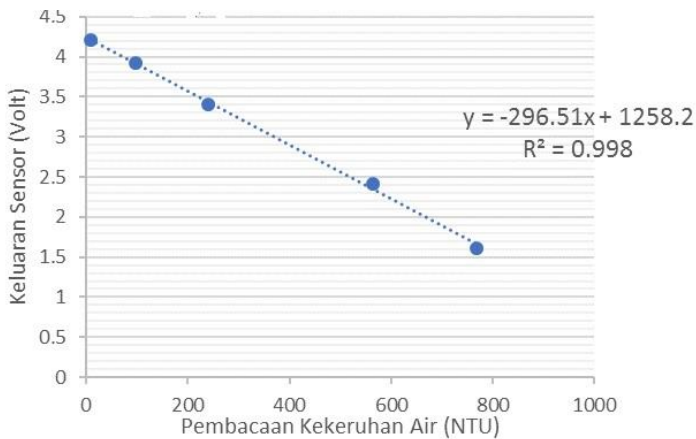
Tabel 4.13 Data Hasil Pengujian *Sensor Kekeruhan*

No	Air yang digunakan	Hasil Pembacaan (NTU)	Parameter Pembanding (NTU)	Hasil Pembacaan (Volt)	Error (%)
1	Air Sungai Keruh	780,81	768	1,6	1,60
2	Air Sungai Semi Keruh	540,65	564	2,42	4,14
3	Air Sungai jernih	250,07	239	3,4	4,42
4	Air Sumur	95,89	97,9	3,92	2,0
5	Air PDAM	9,89	8,49	4,21	14,18
Rata – rata					5,29



Gambar 4.17 Pengujian Kekeruhan Air Menggunakan *Turbidity* Meter

Hasil dari Tabel 4.13 merupakan pengukuran yang dilakukan dengan *Turbidity Meter* seperti pada Gambar 4.17.



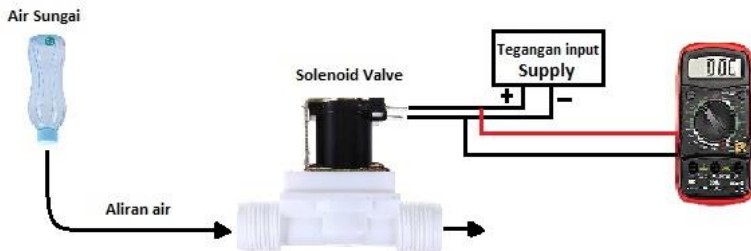
Gambar 4.18 Pengujian Alat Ukur Kekeruhan Air

Data hasil pengujian secara grafik dapat dilihat pada Gambar 4.18. Pada gambar tersebut dapat diketahui nilai regresi linier dari grafik sebesar $y = -296,51x + 1258,2$. Dimana x adalah nilai tegangan yang dihasilkan *sensor* kekeruhan dan y adalah kekeruhan cairan berdasarkan alat ukur acuan. Nilai koefisien determinasi pada regresi linier (r^2) adalah 0,998. Berdasarkan Tabel 4.13, rata-rata nilai *error* pembacaan *sensor* kekeruhan untuk setiap parameter pembacaan perbedaan air sebesar 5.29 %. Air yang dapat dikatakan air bersih memiliki tingkat kekeruhan antara 5 NTU hingga 10 NTU.

4.6 Pengujian Solenoid Valve

Pengujian *Solenoid Valve* merupakan tanggung jawab Rivaldy Hariansyah. Pada pengujian *Solenoid Valve* memiliki maksud dan tujuan yaitu untuk memastikan kinerja dari *Solenoid Valve* dapat bekerja sesuai dengan yang diharapkan.

Pada pengujian pengukuran dilakukan seperti pada Gambar 4.19 dengan menghubungkan *Solenoid Valve* dengan tegangan input 0-17 Volt untuk mengetahui aktif atau tidak dari *Solenoid Valve*. Kemudian *Solenoid Valve* dihubungkan dengan botol yang telah diisi dengan air sungai. Pengujian ini bertujuan untuk mengetahui tegangan kerja dari tegangan *input* yang diberikan pada *Solenoid Valve*. Hasil pengujian ditampilkan pada Tabel 4.14



Gambar 4.19 Pengujian *Solenoid Valve*



Gambar 4.20 Letak *Solenoid Valve* Pada Alat

Pada Gambar 4.20 ditampilkan posisi *Solenoid Valve* pada *prototype* penjernihan air. Pada *prototype* yang dibuat terdapat 5 *Solenoid Valve* yang digunakan.

Tabel 4.14 Data Hasil Pengujian *Solenoid Valve*

No	Tegangan <i>Input</i> (Volt)	Keterangan
1	6	Tidak Aktif
2	8,5	Tidak Aktif
3	10,7	Aktif
4	15,4	Aktif

No	Tegangan <i>Input</i> (Volt)	Keterangan
5	17,7	Aktif

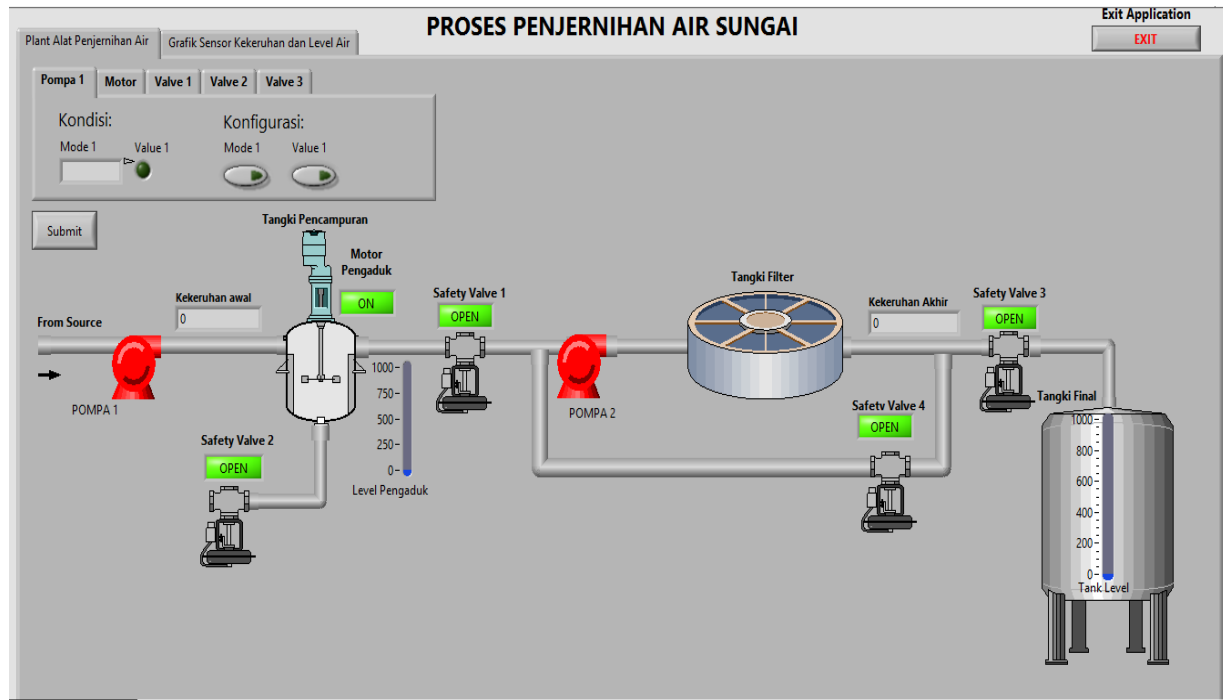
Pada Tabel 4.14 setelah dilakukan pengukuran dari *Solenoid Valve* didapatkan tegangan aktif dari *Solenoid Valve* aktif ketika mendapatkan tegangan 8,5 Volt hingga 17,7 Volt.

4.7 Pengujian HMI LabVIEW

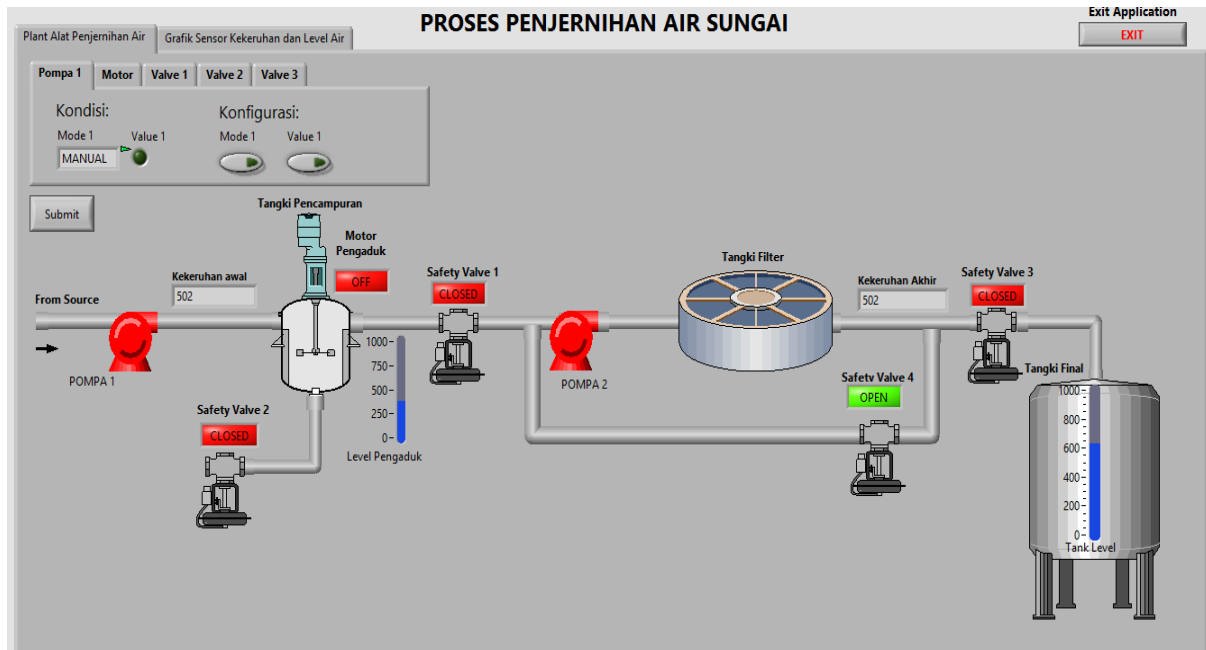
Pengujian HMI LabVIEW merupakan tanggung jawab Rivaldy Hariansyah. Pengujian dilakukan untuk mengetahui *interface* yang digunakan sudah sesuai dengan yang diharapkan atau tidak. Pengujian dilakukan dengan cara melakukan pembacaan data *server* ke LabVIEW dan pengiriman perubahan kondisi aktuator dari LabVIEW. *Flowchart program* HMI LabVIEW dalam mengambil dan menerima data *server* terdapat pada Gambar 3.23. Berikut merupakan cara pengujian pada LabVIEW

1. Hubungkan PC dengan jaringan *internet*
2. Menjalankan Vi pada LabVIEW dengan cara menekan *Run*. Tampilan gambar terdapat pada Gambar 4.21
3. Setelah menekan *Run*, *program* akan secara otomatis mengambil data terakhir dari *webserver*. Tunggu beberapa saat. Tampilan gambar terdapat pada Gambar 4.22
4. Apabila data telah diterima, data tersebut akan ditampilkan di *indicator* pada *program* berupa teks ataupun berupa animasi On/Off. Untuk melakukan perubahan *value* pada aktuator pilih *value* pada setiap aktuator. Apabila tidak menginginkan perubahan *value* pada aktuator tertentu, isi nilai *value* sesuai dengan kondisi saat ini Karena secara *default* data yang dikirimkan bernilai 0 / Off. Apabila sudah dilakukan perubahan klik *submit*
5. Tunggu hingga beberapa saat agar data dapat terkirim

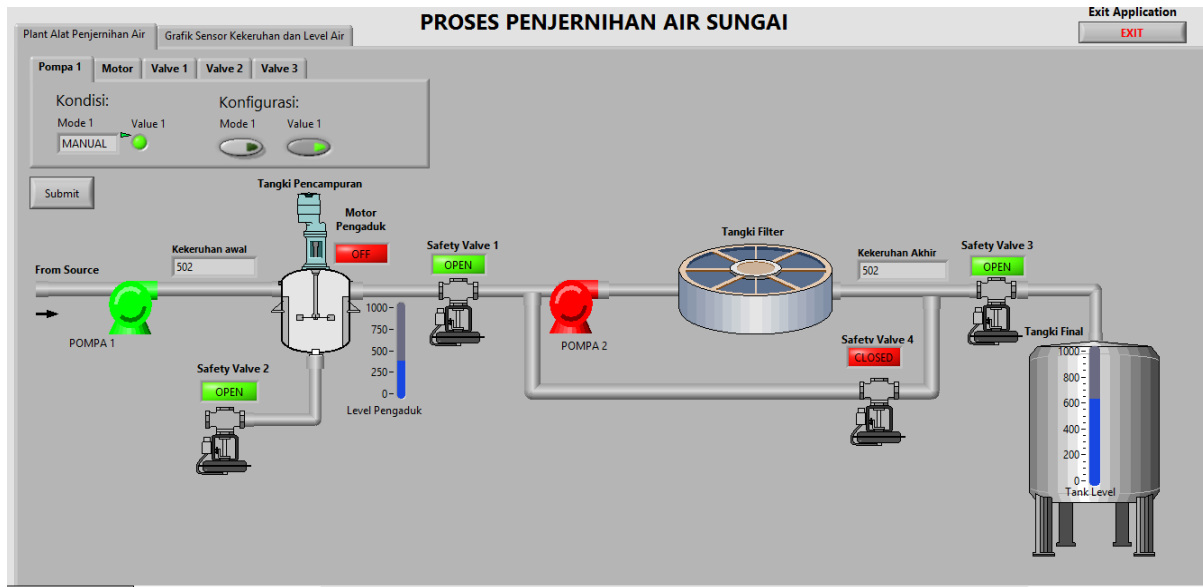
Apabila data telah dikirim maka secara otomatis LabVIEW akan mengambil data *ter-update* dari aktuator sehingga terjadi perubahan pada *indicator* sesuai perintah yang dimasukkan. Tampilan gambar terdapat pada Gambar 4.23



Gambar 4.21 HMI LabVIEW Ketika Di Jalankan



Gambar 4.22 HMI LabVIEW Setelah Mengambil Data Dari Server



Gambar 4.23 HMI LabVIEW Setelah Mengirim Data ke Server

4.8 Pengujian pada Webserver

Pengujian pada *Webserver* merupakan tanggung jawab Rivaldy Hariansyah. Pengujian bertujuan untuk mengetahui *server* yang digunakan dapat menyimpan data yang dikirimkan dari Arduino atau dari aplikasi *client*. Pengujian dilakukan dengan cara memantau data yang diterima oleh *webserver*, mengecek apakah terjadi perubahan yang sesuai dengan keadaan alat, *delay* pengiriman antar data dan pengubahan nilai aktuator. *Flowchart program* pengujian *Webserver* pada Gambar 3.17. Berikut merupakan cara pengujian pada *webserver*:

1. Buka halaman *website* <http://penjernihanair.esy.es/pages/login.php>. Isi *username* dan *password* dengan kata “admin”. Tampilan gambar terdapat pada Gambar 4.24.
2. Klik *submit* dan tunggu hingga *login* ke halaman awal *website*.
3. Pada halaman awal *website* terdapat 4 *Icon* untuk mempermu- dah membuka halaman tertentu pada *website*. Berikut merupakan *icon* beserta fungsi yang terdapat pada Tabel 4.15.

Tabel 4.15 Alamat Halaman *Website* Berdasarkan *Icon*

<i>Icon</i>	Alamat Url Dituju	Fungsi
<i>Sensor Charts</i>	http://penjernihanair.esy.es/pages/history_charts.php	Menampilkan nilai <i>sensor</i> secara grafik <i>realtime</i>
<i>Actuator Tables</i>	http://penjernihanair.esy.es/pages/actuator_tables.php	Menampilkan kondisi <i>actuator</i> berupa tabel. Data yang dit- ampilkan merupakan 100 data terbaru
<i>Sensor Tables</i>	http://penjernihanair.esy.es/pages/sensor_tables.php	Menampilkan kondisi <i>sensor</i> berupa tabel. Data yang dit- ampilkan merupakan 100 data terbaru
<i>Actuator</i>	http://penjernihanair.esy.es/pages/actuator.php	Menampilkan <i>menu</i> untuk meru- bah kondisi dari aktuator serta menampilkan kondisi aktuator berupa grafik <i>realtime</i>

4. Pilih *Sensor Charts*. Maka akan muncul grafik *sensor* kekeru- han dan *sensor level* air. Secara *default* akan ditampilkan kekekeruhan air pada alat. Tampilan gambar terdapat pada Gambar 4.25. Untuk melihat ketinggian air klik pada tab *Level*

Ketinggian Tangki. Tampilan gambar terdapat pada Gambar 4.26.

5. Pilih Aktuator *Table* untuk Menampilkan kondisi aktuator berupa tabel. Tampilan halaman *website* terdapat pada Gambar 4.27. Data yang ditampilkan merupakan 100 data terbaru. Hasil penerimaan data *server* diunduh menjadi sebuah file excel. Data *monitoring* kondisi aktuator dapat dilihat pada Tabel 4.16

Tabel 4.16 Data Excel *Monitoring* Aktuator

<i>created_at</i>	<i>entry_id</i>	<i>Fields</i>					
		1	2	3	4	5	6
2017-05-30 23:27:49	278	0	0	0	0	0	0
2017-05-30 23:28:29	279	0	0	0	0	0	0
2017-05-30 23:30:28	280	0	0	0	0	0	0
2017-05-30 23:32:15	281	0	0	0	0	0	0
2017-05-30 23:33:07	282	0	0	0	0	0	0
2017-05-30 23:33:34	283	0	0	0	0	0	0
2017-05-30 23:34:38	284	0	0	0	0	0	0
2017-05-30 23:38:14	285	0	0	0	0	0	0
2017-05-30 23:38:38	286	0	0	0	0	0	0
2017-05-30 23:39:31	287	0	0	0	0	0	0
2017-05-30 23:39:58	288	0	0	0	0	0	0
2017-05-30 23:40:35	289	0	0	0	0	0	0
2017-05-30 23:41:01	290	0	0	0	0	0	0

Pada Tabel 4.16 ditampilkan 7 buah kolom. Pada kolom pertama merupakan waktu data tersebut diterima *webserver*. Pada kolom *entry id* berisikan no urut data yang diterima. *Field 1* hingga *Field 6* mewakili penomoran dari aktuator untuk mempermudah pengiriman data. Penjelasan mengenai penomoran dapat dilihat pada Tabel 4.17

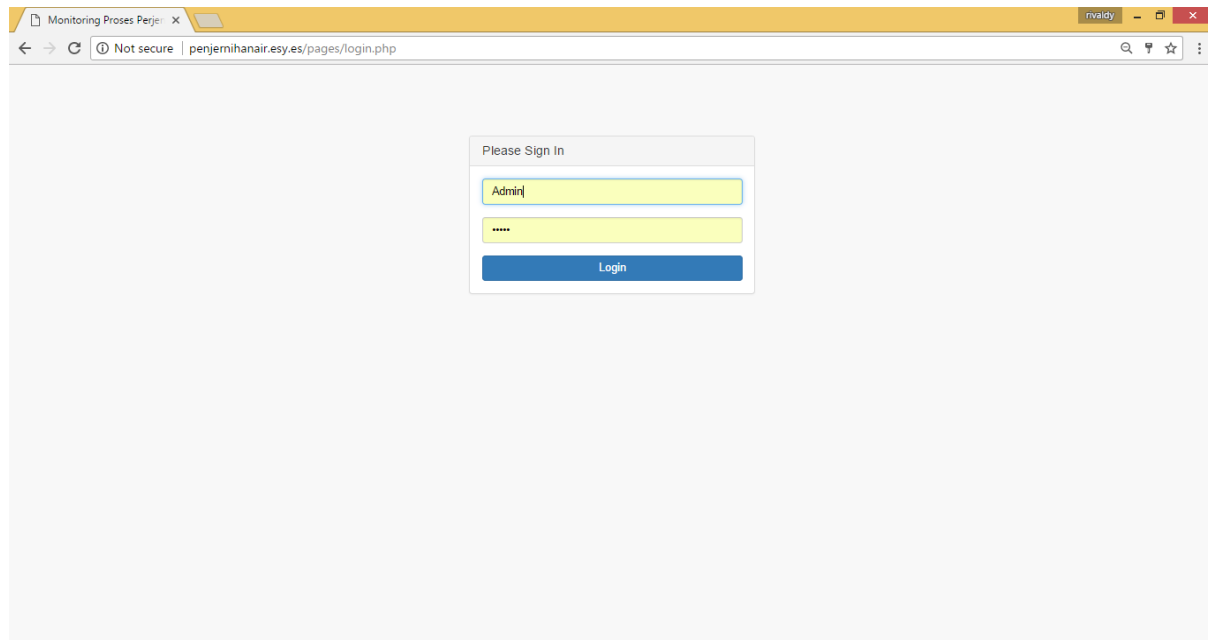
6. Pilih *Sensor Tables*. Pada halaman *website* akan muncul kondisi dari *sensor* sebanyak 100 data terbaru dalam bentuk tabel. Klik pada *download data* untuk men-*download* dalam bentuk file excel. Tampilan gambar terdapat pada Gambar 4.28
7. Pilih Aktuator. Maka akan muncul grafik kondisi aktuator secara *realtime* berdasarkan perubahan waktu. Pada bagian

kontrol dapat diatur agar aktuator berjalan otomatis atau secara manual berdasarkan keinginan pengguna. Tampilan gambar terdapat pada Gambar 4.29

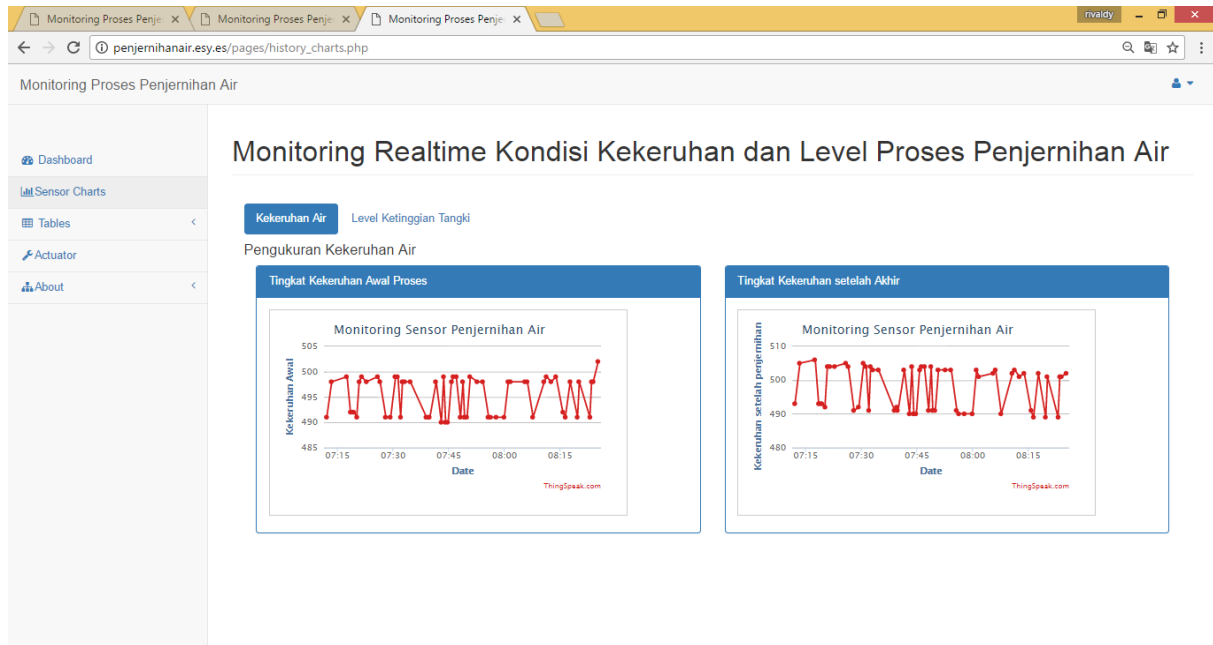
8. Untuk merubah nilai aktuator pilih mode dari aktuator tersebut terlebih dahulu lalu klik *submit* hingga data terkirim. Apabila data terkirim akan muncul respon dari *server* seperti pada Gambar 4.30.
9. Klik Tab Konfigurasi. Pilih kondisi aktuator ON/ OFF. Konfigurasi ini hanya akan muncul apabila memilih *mode* manual. Apabila memilih *mode auto* maka tidak akan muncul pilihan ON/OFF seperti pada Gambar 4.32
10. Apabila tombol *submit* ditekan dan data telah dikirim ke *server*, halaman *website* akan menampilkan pemberitahuan seperti pada Gambar 4.31

Tabel 4.17 Data Penomoran *Field*

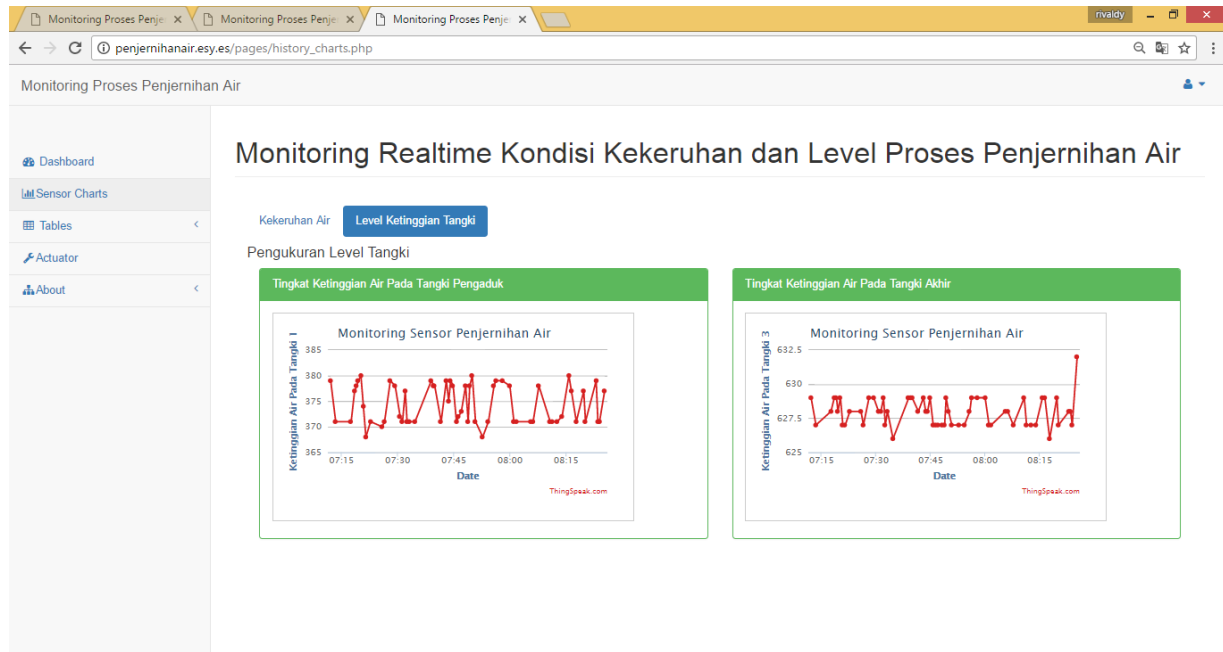
No	Aktuator
1	Pompa 1
2	Pompa 2
3	Motor
4	Valve 1
5	Valve 2
6	Valve 3



Gambar 4.24 *Login Website*



Gambar 4.25 Pengukuran *Sensor Kekeruhan Pada Website*



Gambar 4.26 Pengukuran *Sensor Ketinggian* Pada *Website*

Monitoring Proses Penjernihan Air

Hasil Monitoring Pada Alat Penjernihan Air Sungai

Hasil Monitoring Kondisi Aktuator

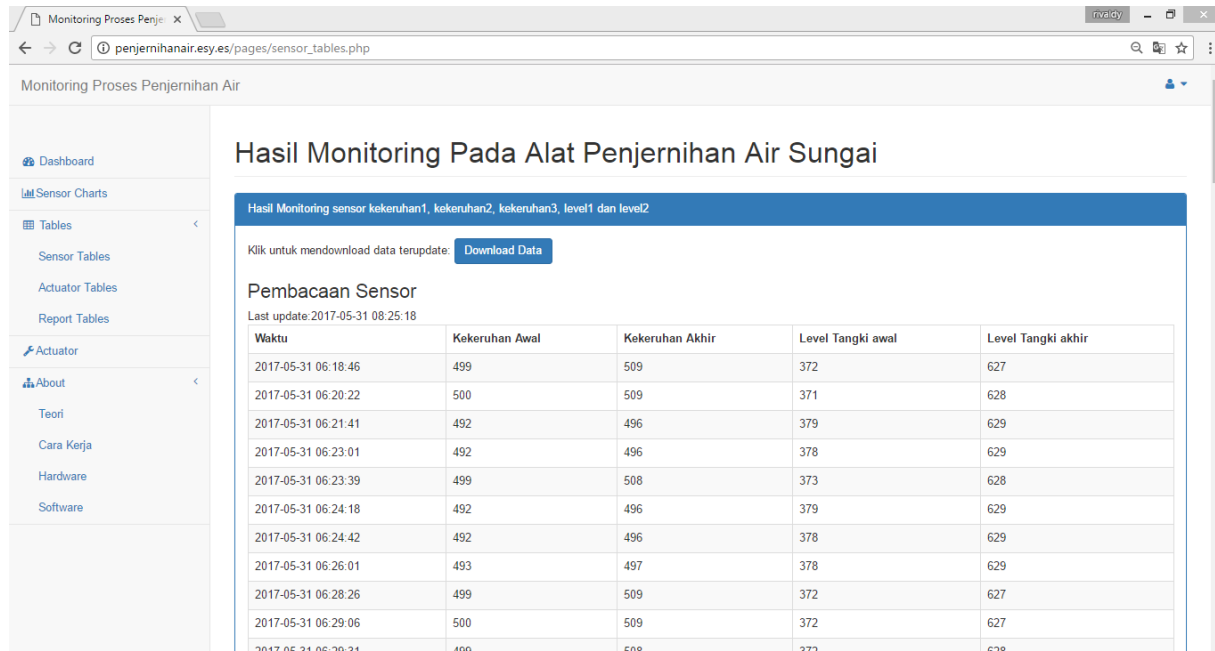
Klik untuk mendownload data terupdate: [Kondisi Aktuator](#) [Mode Aktuator](#)

Kondisi Aktuator

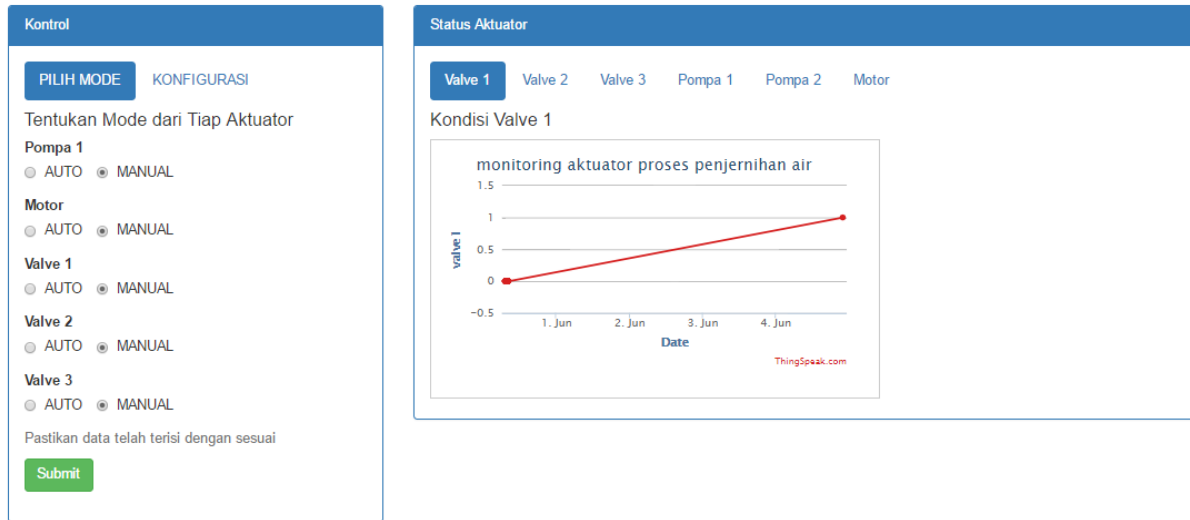
Last update: 2017-06-04 22:08:03

Waktu	Pompa 1	Motor	Valve 1	Valve 2	Valve 3	Motor
2017-05-31 06:27:49	OFF	OFF	OFF	OFF	OFF	OFF
2017-05-31 06:28:29	OFF	OFF	OFF	OFF	OFF	OFF
2017-05-31 06:30:28	OFF	OFF	OFF	OFF	OFF	OFF
2017-05-31 06:32:15	OFF	OFF	OFF	OFF	OFF	OFF
2017-05-31 06:33:07	OFF	OFF	OFF	OFF	OFF	OFF
2017-05-31 06:33:34	OFF	OFF	OFF	OFF	OFF	OFF
2017-05-31 06:34:38	OFF	OFF	OFF	OFF	OFF	OFF
2017-05-31 06:38:14	OFF	OFF	OFF	OFF	OFF	OFF
2017-05-31 06:38:38	OFF	OFF	OFF	OFF	OFF	OFF
2017-05-31 06:39:31	OFF	OFF	OFF	OFF	OFF	OFF
2017-05-31 06:39:58	OFF	OFF	OFF	OFF	OFF	OFF

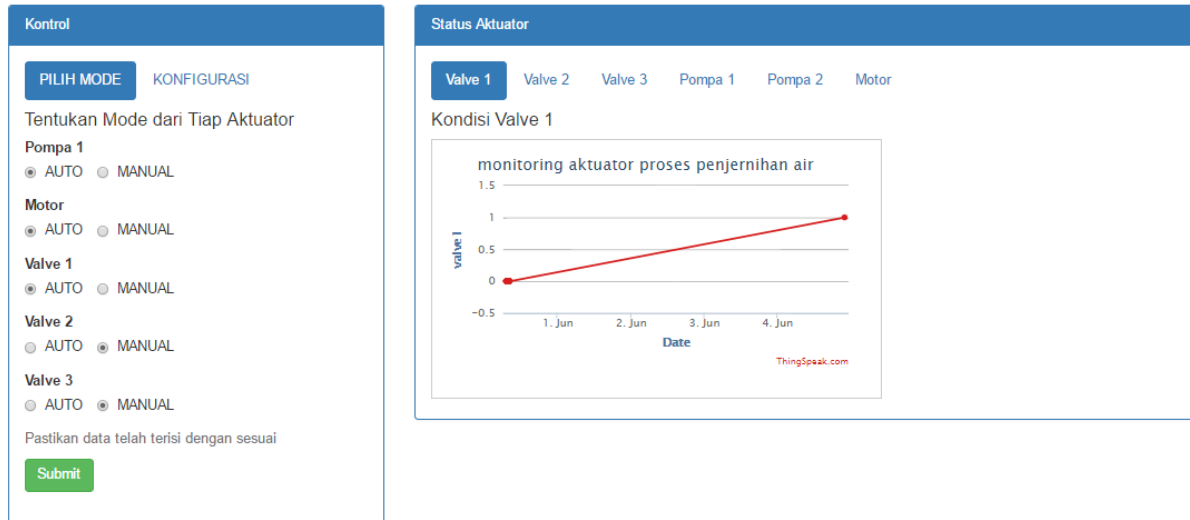
Gambar 4.27 Data Kondisi Aktuator Pada Website



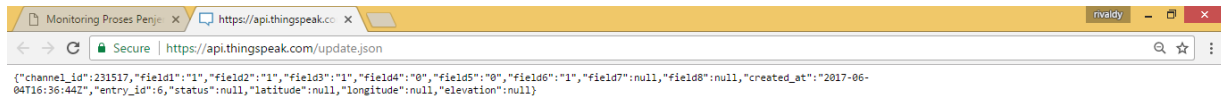
Gambar 4.28 Data Pengukuran *Sensor* Pada *Website*



Gambar 4.29 Kondisi Aktuator



Gambar 4.30 Mengubah Kondisi *Mode* Dari Aktuator



Gambar 4.31 Ketika Data Telah Berhasil Dikirim Ke *Server*

Kontrol

PILIH MODE

KONFIGURASI

Konfigurasi Tiap Aktuator

Pompa 1

Motor

Valve 1

Valve 2

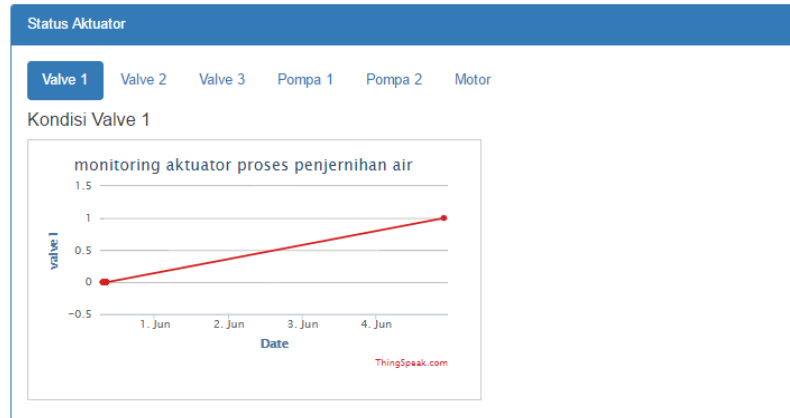
☐ ON ☒ OFF

Valve 3

☐ ON ☒ OFF

Apabila memilih mode AUTO jangan merubah nilai DEFAULT dari Aktuator

Submit



Gambar 4.32 Mengubah Konfigurasi Aktuator

Langkah selanjutnya adalah pengujian pengiriman data ke *web-server Thingspeak.com*. Pengujian dilakukan dengan menggunakan *Ethernet Shield* yang terkoneksi dengan Laptop. Dari hasil percobaan pengiriman data ke *server* diperoleh data pengiriman ke *server* seperti pada Tabel 4.18

Tabel 4.18 Data Pengiriman Ke *Server Thingspeak*

No	Waktu Data Diterima	<i>Delay</i> Pengiriman Data
1	2017-06-05 16:54:29	0
2	2017-06-05 16:54:54	25 detik
3	2017-06-05 16:55:11	17 detik
4	2017-06-05 16:55:26	15 detik
5	2017-06-05 16:55:42	16 detik
6	2017-06-05 16:56:08	26 detik
7	2017-06-05 16:56:35	27 detik
8	2017-06-05 16:56:50	15 detik
9	2017-06-05 16:57:15	25 detik
10	2017-06-05 16:57:42	27 detik
11	2017-06-05 16:58:09	27 detik
12	2017-06-05 16:58:34	25 detik
13	2017-06-05 16:58:59	25 detik

Berdasarkan tabel pengiriman data ke *server* terdapat *delay* antara pengiriman data pertama dengan pengiriman selanjutnya antara 15 detik hingga 27 detik. *Delay* ini dapat disebabkan karena *server Thingspeak.com* membatasi waktu pengiriman hanya dapat dilakukan 15 detik setelah pengiriman sebelumnya.

4.9 Pengujian Aplikasi *Android*

Pengujian aplikasi *Android* merupakan tanggung jawab Achmad Luki Satriawan. Pada pengujian ini dilakukan untuk mengetahui pembacaan data hasil pengukuran yang diakses dari *website* pada aplikasi *Android*. Pengujian dilakukan dengan cara pemantauan data hasil pembacaan *sensor* yang telah dikirim oleh arduino ke *website* kemudian buka aplikasi *Android* yang telah ter-install pada *handphone user* kemudian buka pada *menu* kemudian pada tab *report data*. Pada tab

ini dapat terlihat pembacaan hasil pengukuran *sensor* yang terpasang pada *prototype* yang ditampilkan pada *Android*. Kemudian pengujian tambahan dilakukan dengan mengakses data pembacaan *sensor* dari aplikasi *Android* yang dilakukan pada tempat umum. Berikut ini merupakan pengujian yang dilakukan pada aplikasi *Android*.

Tabel 4.19 Data Pengujian Aplikasi *Android*

No	Tempat	Keterangan	Koneksi
1	Gedung	Didalam mall Tunjungan Plaza lantai 3	Putus-putus
2	Rumah	Jl Gebang Kidul	Lancar
3	Mobil	Berada di dalam kendaraan mobil	Lancar
4	Lapangan Bola	Lapangan ITS	Lancar
5	Taman Bungkul	Taman Bungkul Surabaya	Lancar
6	Stasiun Kereta	Stasiun Kereta Api Pasar Turi Surabaya	Lancar

Pada Tabel 4.19 merupakan pengujian yang dilakukan. Saat proses pemantauan dilakukan yaitu menggunakan *Android* diperlukan koneksi *internet* yang stabil. Hal ini dikarenakan akan berpengaruh pada proses pemantauan pada perangkat yang bersangkutan. Pada proses pemantauan yang dilakukan pada perangkat *handphone user* koneksi akan mengalami putus-putus saat berada di tempat gedung dan akan lancar disaat berada pada tempat yang terbuka. Terdapat grafik yang menggambarkan pengukuran dari masing-masing *sensor* seperti Gambar 4.33 yang merupakan hasil pembacaan *sensor* yang ditampilkan secara grafik dan *report data* hasil pengukuran seperti pada Gambar 4.34. Data hasil pembacaan tersebut didapat dari pengambilan data secara *realtime*.



Gambar 4.33 Data *Sensor* Di *Android*.

Pada Gambar 4.33 merupakan data pembacaan *sensor turbidity* yang terukur pada *sensor*. Hasil pembacaan *sensor* yang dikirim ke *clouds* kemudian akan diolah berupa grafik oleh *Thingspeak.com*

Waktu	Level Awal	Level Akhir
2017-06-05 16:54:29	1022	1022
2017-06-05 16:54:54	1022	1022
2017-06-05 16:55:11	1022	1022
2017-06-05 16:55:26	1022	1022

Gambar 4.34 *Report Data* Pengukuran *Sensor* Di *Android*

Pada Gambar 4.34 merupakan hasil pengujian yang dilakukan pada tab *report data* yang diakses melalui *handphone user*. Pada tahap ini dapat terlihat kondisi *level* dari air yang ditampung dari kondisi awal hingga kondisi akhir dari air yang ditampung.

4.10 Pengujian Proses Penjernihan Air

Pengujian proses penjernihan air merupakan tanggung jawab Rivaldy Hariansyah. Pengujian proses penjernihan air bertujuan untuk mempraktekkan studi literatur mengenai proses penjernihan air dari beberapa buku penjernihan air dan Tugas Akhir yang terkait dengan proses penjernihan air. Langkah – Langkah pengujian proses penjernihan air yaitu menyiapkan air Sungai Jagir sebanyak 6x botol 1,5 Liter air mineral. Selanjutnya Menyiapkan alat filter dan wadah pencampuran tawas. Tuangkan air sungai pada wadah pencampuran tawas. Masukkan tawas yang telah dicairkan dengan konsentrasi 80% kedalam wadah pencampuran. Aduk dengan cepat untuk mencampurkan air dan tawas secara merata selama 1 menit. Pada Gambar 4.36 merupakan kondisi air sungai setelah dicampur dengan Tawas. Terbentuk gumpalan gumpalan padat yang merupakan campuran kotoran didalam air yang saling terikat. Terlihat kondisi air kini mulai terlihat lebih jernih. Tunggu selama 30 detik, campurkan larutan polimer. Aduk secara perlahan untuk mempercepat proses penggumpalkan kotoran. Tuangkan air pada filter secara perlahan agar tidak terjadi tumpah. Filter yang dibuat terdiri dari 3 bagian yaitu bagian pertama berisi kapas yang akan menyaring floks besar yang terlihat oleh mata. Pada bagian kedua adalah arang kelapa, berfungsi untuk menghilangkan bau tidak sedap didalam air dan yang ketiga merupakan Batu Zeolit atau biasa disebut batu akuarium berfungsi untuk menyerap kotoran kecil yang tidak kasat mata melalui pori – porinya. Lakukan proses filtrasi selama 3x dan ambil *sample* produk setiap prosesnya. Terakhir endapkan *sample* produk selama 1 hari untuk menghasilkan air yang jernih. Hasil dari pengujian proses penjernihan air didapatkan data sebagai berikut.

Tabel 4.20 Hasil Pengujian Penjernihan Air

No	Banyak Filtrasi	kekeruhan Air	Bau
1	Tanpa Filtrasi	Keruh	Berbau
2	1 kali	Sedang	Berbau
3	2 kali	Jernih	Berbau
4	3 kali	Jernih	Tidak Berbau

Berdasarkan data pengujian pada Tabel 4.20 diperlukan filtrasi minimal sebanyak 3x lalu dilakukan pengendapan selama 1 hingga 2 hari. Pada Gambar 4.35 Merupakan kondisi air setelah diendapkan

selama 2 hari. Pada gelas ukur ke-1 merupakan kondisi air sebelum dilakukan proses filtrasi. Terdapat lumut pada bagian bawah tabung menunjukkan bahwa masih terdapat zat organik didalamnya. Pada gelas ukur ke-2 dan gelas ukur ke-3 tidak terlihat perbedaan mencolok dari gambar tersebut karena masih melalui filtrasi 1x dan 2x. Terdapat banyak kotoran yang mengendap dan ukurannya besar, hal ini dapat terjadi karena proses penyaringan belum maksimal sehingga floks dari proses koagulasi dan flokulasi masih tertinggal didalam air dan mengendap. Pada gelas ukur ke-4 masih terdapat endapan pada tabung, ukurannya kecil dan lembut menunjukkan proses filtrasi telah menghilangkan floks besar. Sisa floks ini tidak dapat dihilangkan hanya dengan filter biasa



Gambar 4.35 Kondisi Air Setelah Di Endapkan Selama 2 Hari



Gambar 4.36 Kondisi Air Setelah Dicampurkan dengan Tawas

4.11 Pengujian Alat Secara Keseluruhan

Pengujian ini dilakukan untuk menunjukkan alat telah berjalan tanpa gangguan dan sesuai dengan fungsinya. Rancangan rangkaian elektronika alat pada Gambar 3.4 dan bentuk alat secara keseluruhan pada Gambar 4.37.

Proses pengujian alat secara keseluruhan melalui beberapa tahapan.

1. Alat dihubungkan dengan sumber listrik AC 220 Volt.
2. Hubungkan kabel RJ-45 dengan Laptop dan kabel USB. Gambar pada Lampiran B.5 menunjukkan letak kabel AC pada nomor 1, USB pada nomor 2 dan RJ-45 pada nomor 3.
3. Lakukan proses *sharing internet* dari Laptop ke Arduino untuk memberikan Arduino akses ke *internet*
4. Buka *Serial Monitor* Arduino. Tunggu hingga proses *sharing* berhasil dan Arduino dapat mengirim data ke *server*.
5. Aktifkan saklar pada kontroler untuk memberikan *supply* listrik pada aktuator. Gambar pada Lampiran B.5 menunjukkan Letak Saklar pada kontak kontroler
6. Pompa Air 1 akan menyala hingga *level* air mencapai 25 Cm. Gambar proses pengisian Tangki terdapat pada Lampiran B.8
7. *Sensor level* akan naik seiring dengan kenaikan *volume* air pada tangki penampungan awal. Detail pengukuran *Sensor Level* dilakukan seperti Pengujian *Sensor* Ketinggian pada Sub Bab 4.2. Alur aliran Air dari wadah air sungai ke Tangki penampungan awal terletak pada Lampiran B.3
8. *Sensor* kekeruhan akan membaca tingkat kekeruhan air sungai yang melalui pipa menuju tangki penampungan awal. *Sensor* kekeruhan awal merupakan hasil pengukuran kekeruhan air yang melewati pipa menuju tangki penampungan awal. *Sensor* kekeruhan akhir merupakan pembacaan *sensor* kekeruhan yang telah melalui filter air. Detail pengukuran *sensor* kekeruhan dilakukan seperti Pengujian *Sensor* Kekeruhan pada Sub Bab 4.5.
9. Hasil pembacaan *level* dan kekeruhan pada awal proses akan diolah Arduino untuk dikirim ke *webserver*.
10. Apabila tangki penampungan awal penuh maka pompa air akan otomatis nonaktif.
11. Tuangkan cairan tawas 10ml pada tangki penampungan.

12. Tekan *push button* 1x dan motor akan mengaduk selama 1 menit. Letak *push button* pada kotak kontroler ditunjukkan pada Lampiran B.3
13. Air akan keluar melalui *Valve 1* menuju Pompa 2
14. Pompa 2 akan memompa air ke akuarium filter. Apabila *sensor level 1* terdeteksi air maka Pompa 2 dan *Valve 1* akan nonaktif sehingga pada proses filter, air tidak akan meluber. Gambar proses pengisian filter terdapat Lampiran B.8
15. Proses pengisian tangki penampungan awal akan berlangsung secara terus menerus seperti pada tahap nomor 1 hingga 14.
16. Air yang ada dalam filter akan melalui beberapa tahapan penyaringan yaitu kapas, karbon arang, Batu Zeolit, pasir dan kapas serta proses sedimentasi pada akhir filter. Alur aliran air dari pompa ke-2 hingga keluar dari tahap filtrasi terdapat pada Lampiran B.4
17. Pada akhir filter *sensor level* Filter 2 akan aktif maka *Valve 3* dan *Valve 4* akan membuka dan menutup sesuai pembacaan *sensor* kekeruhan
18. *Sensor* kekeruhan pada akhir proses filtrasi berfungsi untuk membaca tingkat kekeruhan air, apabila tingkat kekeruhan air diatas 100 NTU maka *Valve 2* akan terbuka dan *Valve 3* akan menutup dan Air akan dikembalikan ke Pompa 2 tetapi apabila tingkat kekeruhan air berada dibawah 100 NTU maka *Valve 2* akan menutup dan *Valve 3* akan membuka sehingga air akan masuk ke tangki penampungan akhir. Tingkat kekeruhan air dan ketinggian air pada tangki penampungan akhir akan diolah oleh Arduino dan dikirim ke *webserver*.



Gambar 4.37 Alat Telemetri Proses Penjernihan Air

Tabel 4.21 Hasil Pengujian Pembacaan *Sensor* Pada Alat

Waktu	<i>Sensor Kekeruhan (NTU)</i>		<i>Sensor Ketinggian (Cm)</i>	
	Awal	Akhir	Awal)	Akhir
09.00	160	155	0	0
09.01	850	150	12	0
09.02	870	155	20	0
09.03	869	154	25	0
09.04	865	154	25	0
09.05	867	155	25	0
09.06	868	155	17	0
09.07	868	154	9	0
09.08	870	155	0	0
09.09	850	154	11	0
09.10	865	154	19	0
09.11	864	152	25	0
09.12	865	151	25	0
09.13	864	150	25	0
09.14	865	90	19	0
09.15	865	90	8	0
09.16	855	89	0	0
09.17	860	90	0	0
09.18	867	90	13	1
09.19	865	89	19	2
09.20	866	90	25	4

Waktu	Sensor Kekeruhan (NTU)		Sensor Ketinggian (Cm)	
	Awal	Akhir	Awal	Akhir
09.21	865	90	25	5
09.22	865	90	25	6
09.23	865	90	19	7
09.24	866	90	6	9
09.25	865	91	0	11
09.26	870	90	0	12
09.27	867	90	13	13
09.28	865	90	18	14
09.29	867	90	25	15
09.30	865	90	25	17
09.31	868	89	25	18
09.32	865	90	19	19
09.33	865	90	10	20
09.34	866	90	0	21
09.35	865	90	0	22
09.36	865	89	13	23
09.37	865	91	18	24
09.38	155	154	25	25

Tabel 4.22 Hasil Pengujian Kondisi Aktuator Pada Alat

Waktu	Aktuator					
	1	2	3	4	5	6
09.00	ON	OFF	OFF	OFF	OFF	OFF
09.01	ON	OFF	OFF	OFF	OFF	OFF
09.02	ON	OFF	OFF	OFF	OFF	OFF
09.03	OFF	OFF	OFF	OFF	OFF	OFF
09.04	OFF	ON	OFF	OFF	OFF	OFF
09.05	OFF	ON	OFF	OFF	OFF	OFF
09.06	OFF	OFF	ON	ON	OFF	OFF
09.07	OFF	OFF	ON	ON	OFF	OFF
09.08	ON	OFF	OFF	OFF	OFF	OFF
09.09	ON	OFF	OFF	OFF	OFF	OFF
09.10	ON	OFF	OFF	OFF	OFF	OFF
09.11	OFF	OFF	OFF	OFF	OFF	OFF
09.12	OFF	ON	OFF	OFF	OFF	OFF
09.13	OFF	ON	OFF	OFF	OFF	OFF
09.14	OFF	OFF	ON	ON	OFF	OFF
09.15	OFF	OFF	ON	ON	OFF	OFF

Waktu	Aktuator					
	1	2	3	4	5	6
09.16	ON	OFF	OFF	OFF	OFF	OFF
09.17	ON	OFF	OFF	OFF	ON	ON
09.18	ON	OFF	OFF	OFF	ON	OFF
09.19	ON	OFF	OFF	OFF	ON	OFF
09.20	OFF	OFF	OFF	OFF	ON	OFF
09.21	OFF	ON	OFF	OFF	ON	OFF
09.22	OFF	ON	OFF	OFF	ON	OFF
09.23	OFF	OFF	ON	ON	ON	OFF
09.24	OFF	OFF	ON	ON	ON	OFF
09.25	ON	OFF	OFF	OFF	ON	OFF
09.26	ON	OFF	OFF	OFF	ON	OFF
09.27	ON	OFF	OFF	OFF	ON	OFF
09.28	ON	OFF	OFF	OFF	ON	OFF
09.29	OFF	OFF	OFF	OFF	ON	OFF
09.30	OFF	ON	OFF	OFF	ON	OFF
09.31	OFF	ON	OFF	OFF	ON	OFF
09.32	OFF	OFF	ON	ON	ON	OFF
09.33	OFF	OFF	ON	ON	ON	OFF
09.34	ON	OFF	OFF	OFF	ON	OFF
09.35	ON	OFF	OFF	OFF	ON	ON
09.36	ON	OFF	OFF	OFF	ON	OFF
09.37	ON	OFF	OFF	OFF	ON	OFF
09.38	OFF	OFF	OFF	OFF	OFF	OFF

Keterangan Pada Aktuator:

No. 1 : Pompa Air 1

No. 2 : Motor

No. 3 : Pompa Air 2

No. 4 : *Valve 1*

No. 5 : *Valve 2*

No. 6 : *Valve 4*

Pada Tabel 4.21 dan Tabel 4.22 ditunjukkan mengenai kondisi alat bekerja selama 38 menit mulai pukul 09.00 hingga 09.38. Pengujian dilakukan dari tangki penampungan akhir belum terisi air hingga terisi penuh. Proses Pengisian Tangki penampungan awal dilakukan selama 2 menit dari posisi 0 Cm hingga 25 Cm. Proses pengisian Filter dilakukan selama 2 hingga 3 menit hingga tangki penampungan awal terkuras habis dan melakukan pengisian kembali. Tangki penampungan akhir mulai terisi pada menit ke 09.18. *Valve 4* akan melakukan pembersihan tangki

penampungan awal setiap 2x proses pencampuran dengan tawas untuk menghilangkan endapan pada tangki.

Pada pengujian tersebut terdapat beberapa kendala yang terjadi antara lain:

- 1 Koneksi *internet* yang tidak stabil membuat proses *looping program* menjadi terganggu sehingga terkadang terputus disaat proses berjalan.
- 2 *Delay server thingspeak.com* dalam menerima data membutuhkan 15 hingga 20 detik sehingga tidak setiap perubahan terekam.
- 3 Filter Air perlu dilakukan pembersihan kembali setelah tangki penampungan akhir penuh sehingga diperlukan beberapa kali perawatan pada alat apabila digunakan pada jangka panjang.
- 4 Bahan kimia yang dilarutkan pada proses penjernihan air seringkali menghambat pada aliran *valve* sehingga pada saat *valve* bekerja membuka atau menutup tidak bekerja 100% .

-----Halaman ini sengaja dikosongkan-----

BAB V

PENUTUP

Dari hasil yang telah didapatkan selama proses perancangan dan pembuatan serta proses pengujian dan analisa data untuk Tugas Akhir ini, maka dapat diambil beberapa kesimpulan dan saran yang berguna untuk perbaikan dan pengembangan agar nantinya bermanfaat.

5.1 Kesimpulan

1. Dalam hasil pengambilan data ketinggian air berdasarkan Tabel 4.2 rata-rata penurunan tegangan setiap 1 Cm 2,74 Volt dan pada Tabel 4.3 rata-rata penurunan tegangan setiap 1 Cm sebesar 2,80Volt
2. Dalam hasil pengambilan data pada *sensor* kekeruhan mempunyai *error* sebesar 5,29% dan hasil pengukuran menunjukkan air bersih memiliki tingkat kekeruhan antara 5 hingga 10 NTU dan tegangan *sensor* sebesar 4,21 Volt.
3. Pada pengujian aplikasi *Android* pengiriman dan penerimaan data terganggu pada lingkungan yang terdapat gedung tinggi
4. *Database server* menggunakan *Thingspeak.com* memiliki *delay* pengiriman data sebesar 15 detik hingga 27 detik.

5.2 Saran

Saran untuk penelitian selanjutnya adalah dapat menggunakan akun *premium* pada *Thingspeak.com* sehingga pengiriman dapat dilakukan lebih cepat tanpa ada batasan *delay* pengiriman. Pada pemberian tawas dapat dibuat otomatis sehingga *monitoring* alat lebih efektif. Pada pengujian kekeruhan air menggunakan *sensor* kekeruhan ataupun *Turbidity meter* dilakukan pada tempat dan waktu yang bersamaan.

-----Halaman ini sengaja dikosongkan-----

DAFTAR PUSTAKA

- [1] Masduqi Ali, Assomadi, **Operasi & Proses Pengolahan Air**, ITSPress : Surabaya, 2012.
- [2] Abhimanyu Mathur , **Insight - Learn the Working of a Turbidity Sensor** , URL: <https://www.engineersgarage.com/insight/how-turbidity-sensor-works>, 30 Mei 2017
- [3] Antoni Susiono, “Aplikasi SCADA System pada Miniatur Water Level Control”, **Tugas Akhir** , Teknik Elektro, Universitas Kristen Petra, Surabaya, 2006
- [4] Artanto, Dian, **Interaksi Arduino dan Labview**, Elex Media Komputindo: Jakarta, 2012.
- [5] Prasetio, Adi, **Buku Pintar Webmaster Untuk Pemula**, Mediakita: Jakarta, 2015.
- [6] Istiyanto, Jazi Eko, **Pengantar Elektronika dan Instrumentasi dengan Pendekatan Project Arduino + Android**, Penerbit Andi: Yogyakarta, 2014
- [7] Adi Saputra, Rancang Bangun Pengontrolan Daya Listrik Menggunakan Relay Berbasis Mikrokontroler ATmega8532, **Tugas Akhir**, Teknik Elektro, Universitas Islam Negeri Sultan Syarif Kasim: Pekanbaru, 2013
- [8] Kumaidi Agus, **Materi Mata Kuliah Telemetry dengan Basic Ilmu Android, Mikrokontroler, Komunikasi dan Jaringan**, URL: <http://www.ilmuotomasi.com/2017/03/android-materi-mata-kuliah-telemetry.html>, 5 Mei 2017
- [9] Pratama, I Putu Agus Eka, **Wireless Sensor Network**, Informatika: Bandung, 2015.
- [10], **Learn More about ThingSpeak**, URL: https://thingspeak.com/pages/learn_more, 17 Mei 2017
- [11], **Arduino Ethernet Shield-R3**, URL: <http://www.famosastudio.com/ethernet-shield>, 2 Juni 2017
- [12], **Buck Converter**, URL: <http://www.sandielektronik.com/2016/01/buck-converter.html>, 7 Juni 2017

- [13] Deni Bram,Toto Haryanto, **Apa Sih App Inventor Itu**, URL:
<https://www.totoharyanto.com/apa-sih-app-inventor-itu/> , 2 Juni
2017, 8 Juni 2017
- [14] **DC Motor 12V** , Ebay,November 2008
- [15]**12V Electric Plastic Solenoid Valve** ,
electricsolenoidvalves, September 2016
- [16] **Rotary Wirewound** , AliExpress,November 2008

LAMPIRAN A

A.1 Lampiran Program Keseluruhan

```
#include <SPI.h>
#include <Ethernet.h>
#define VOLTAGE_MAX 100.0
#define VOLTAGE_MAXCOUNTS 1023.0
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED};
EthernetClient client;
const int aktuatorPin[] = {2,3,4,5,6,7,8}; //2,3,4,5,6,7,8 aktuator
const int sensorPin[] = {A0,A1,A2,A3}; //Pin pada sensor analog
tangki1,tangki2,tangki3,kekeruhan1,kekeruhan2,kekeruhan3
const int buttonPin = 9;
const int levelPin1= A5;
const int levelPin2= A4;
int valve1=6;
int valve2=5;
int valve3=4;
int valve4=3;
int motor=0;
int pompa1=1;
int pompa2=2;
int buttonState = 0;
int level1State = 0;
int level2State = 0;
int Tangki1State = 0;
int counter_tangki=0;
int sensorValue[] = {0,0,0,0};
unsigned long pengadukan-
Time,pembersihanTime,aksesWebTime;
char c;
char dataStr[150]; //variabel penyimpanan pembacaan string
char server1[] = "/channels/229938/feeds.json?results=1
HTTP/1.1";
char server2[] = "/channels/231517/feeds.json?results=1
HTTP/1.1";
char server_sensor[] = "/update?api_key=
MJJLLLZYBZ2MH2F5&field1="
```

```

char server_kirim[] =
"/update?api_key=MYRNJKUPL5WIS7HQ&field1="; //pengiriman
aktuator
    int x = 0;
    int y=0;
    int aktuator[]={0,0,0,0,0,0}; //variabel menyimpan aktuator
    int mode[]={1,1,1,1,1,1}; //variabel menyimpan mode aktuator
    int buffer;
    int temp=0;

    void setup() {
        Serial.begin(9600);
        //Ethernet.begin(mac);
        pinMode(buttonPin, INPUT);
        pinMode(levelPin1, INPUT);
        pinMode(levelPin2, INPUT);
        for(int i=0;i<=6;i++){
            pinMode(aktuatorPin[i],OUTPUT);
        }
        aksesWebTime = millis();
        for(int i=0;i<=6;i++){
            digitalWrite(aktuatorPin[i],HIGH);
        }
    }

    void loop() {
        for(int s=0;s<=3;s++){
            sensorValue[s] = analogRead(sensorPin[s]);
            //(VOLTAGE_MAX / VOLTAGE_MAXCOUNTS);
            delay(10);
            Serial.print("sensorValue");
            Serial.print(s);
            Serial.print(": ");
            Serial.println(sensorValue[s]);
        }
        level1State = digitalRead(levelPin1);
        level2State = digitalRead(levelPin2);
        Serial.print("tangki 1 State:");
        Serial.println(Tangki 1 State);
    }

```

```

        if(sensorValue[1]>=985){ //610 adalah nilai penuh dari tangki2
untuk menjaga agar tidak banjir
            if(sensorValue[0]<=985&&mode[pompa1]==0){ //menjaga agar
tangki1 tidak banjir
                digitalWrite(aktuatorPin[pompa1],HIGH);
                aktuator[pompa1]=0;
            }
            if(level1State==HIGH&&mode[valve1]==0){//menjaga agar filter
tidak penuh
                digitalWrite(aktuatorPin[valve1],HIGH);
                aktuator[3]=0;
            }
            if(Tangki1State==0){
                if((millis() - pembersihanTime) > 10000){
                    if(mode[valve3]==1){
                        digitalWrite(aktuatorPin[valve3],HIGH);
                    }
                    if(mode[pompa1]==1){
                        digitalWrite(aktuatorPin[pompa1],LOW);
                        aktuator[pompa1]=1;
                    }
                    if(mode[valve1]==1){
                        digitalWrite(aktuatorPin[valve1],HIGH);
                        aktuator[valve1]=0;
                    }
                }
                if(sensorValue[0]<=985){
                    counter_tangki=counter_tangki+1;
                    Tangki1State=1;
                }
            }
        }
        if(Tangki1State==1){
            buttonState = digitalRead(buttonPin);
            if(mode[pompa1]==1||sensorValue[0]<=985){
                digitalWrite(aktuatorPin[pompa1],HIGH);
                aktuator[pompa1]=0;
            }
            if(buttonState == HIGH) {
                if(mode[motor]==1){
                    digitalWrite(aktuatorPin[motor],LOW);

```

```

    aktuator[motor]=1;
    pengadukanTime = millis();
  }
  Tangki1State=2;
}
}
if(Tangki1State==2){
  if((millis() - pengadukanTime) > 5000){
    if(mode[motor]==1){
      digitalWrite(aktuatorPin[motor],HIGH);
      aktuator[motor]=0;
    }
    if(sensorValue[0]>=985&&level1State==LOW){
      if(mode[valve1]==1){
        digitalWrite(aktuatorPin[valve1],LOW);
        aktuator[valve1]=1;
      }
    }
  }
  if(sensorValue[0]>=985&&level1State==HIGH){
    if(mode[valve1]==1){
      digitalWrite(aktuatorPin[valve1],HIGH);
      aktuator[valve1]=0;
    }
  }
}
if(sensorValue[0]>=1000){ //nilai bawah dari tangki 1 385
  if(counter_tangki%2==0){
    if(mode[valve3]==1){
      digitalWrite(aktuatorPin[valve3],LOW);
      pembersihanTime=millis();
    }
  }
  else{
    if(mode[valve3]==1){
      digitalWrite(aktuatorPin[valve3],HIGH);
    }
  }
  Tangki1State=0;
}
}

```

```

if(sensorValue[3]<=500){
    if(mode[valve2]==1){
        digitalWrite(aktuatorPin[valve2],LOW);
        aktuator[valve2]=1;}
    if(mode[valve4]==1){
        digitalWrite(aktuatorPin[valve4],HIGH);
        aktuator[valve4]=0;
    }
}
else{
    if(mode[valve2]==1){
        digitalWrite(aktuatorPin[valve2],HIGH);
        aktuator[valve2]=0;}
    if(mode[valve4]==1){
        digitalWrite(aktuatorPin[valve4],LOW);
        aktuator[valve4]=1;
    }
}

if(aktuator[valve1]==1||(level2State==HIGH
&&aktuator[valve2]==0)){
    digitalWrite(aktuatorPin[pompa2],LOW);
    aktuator[pompa2]=1;
}
else{
    digitalWrite(aktuatorPin[pompa2],HIGH);
    aktuator[pompa2]=0;
}
}
else{
    for(int off=0;off<=5;off++){
        digitalWrite(aktuatorPin[off],HIGH);
        aktuator[off]=0;
    }
}
/*
    if((millis() - aksesWebTime) > 1000){ //menentukan interval
akses web server
    if(client.connect("api.thingspeak.com", 80)) {
        // Make a HTTP request:

```

```

client.print("GET ");
if(y%4==1){
    client.println(server1);
    buffer=1;
}
if(y%4==2){
    client.println(server2);
    buffer=2;
}
if(y%4==3){
    client.print(server_sensor);
    client.print(sensorValue[0]);
    client.print("&field2=");
    client.print(sensorValue[1]);
    client.print("&field3=");
    client.print(sensorValue[2]);
    client.print("&field4=");
    client.print(sensorValue[3]);
    client.println(" HTTP/1.1");
}
else{
    client.print(server_kirim);
    client.print(aktuator[pompa1]);
    client.print("&field2=");
    client.print(aktuator[motor]);
    client.print("&field3=");
    client.print(aktuator[valve1]);
    client.print("&field4=");
    client.print(aktuator[valve2]);
    client.print("&field5=");
    client.print(aktuator[valve3]);
    client.print("&field6=");
    client.print(aktuator[pompa2]);
    client.println(" HTTP/1.1");
}
client.println("Host: api.thingspeak.com");
client.println("Connection: close");
client.println();
y=y+1;
}

```

```

x=0;
while(client.available()) { //membaca jawaban dari server
    c=client.read();
    if(c=='['){
        while(c!=']'){
            c=client.read();
            dataStr[x]=c;
            Serial.print(dataStr[x]);
            x++;
        }
    }
}
if(!client.connected()) {
    const char delim[] = "{}\":,.";
    char *firstItem;
    char *temp;
    char *data1;
    char *data2;
    char *data3;
    char *data4;
    char *data5;
    char *data6;
    char *data7;
    firstItem = strtok(dataStr,delim);
    for(int i=0;i<=5;i++){
        temp = strtok(NULL,delim);
    }
    data1=strtok(NULL,delim);
    temp = strtok(NULL,delim);
    data2=strtok(NULL,delim);
    temp = strtok(NULL,delim);
    data3=strtok(NULL,delim);
    temp = strtok(NULL,delim);
    data4=strtok(NULL,delim);
    temp = strtok(NULL,delim);
    data5=strtok(NULL,delim);
    Serial.println("Sixth item in string:");
    if(strcmp(data1,"") != 0) //apabila data1 tidak bernilai 0
    {
        if(buffer==2){

```

```

Serial.println("Mode");
Serial.println(data1);
if(strcmp(data1,"1") == 0)
    mode[pompa1]=1;
else
    mode[pompa1]=0;
Serial.println(data2);
if(strcmp(data2,"1") == 0)
    mode[motor]=1;
else
    mode[motor]=0;
Serial.println(data3);
if(strcmp(data3,"1") == 0)
    mode[valve1]=1;
else
    mode[valve1]=0;
Serial.println(data4);
if(strcmp(data4,"1") == 0)
    mode[valve2]=1;
else
    mode[valve2]=0;
Serial.println(data5);
if(strcmp(data5,"1") == 0)
    mode[valve3]=1;
else
    mode[valve3]=0;
buffer=0;
}
if(buffer==1){
    Serial.println("KONFIGURASI");
    Serial.println(data1);
    if(mode[pompa1]==0){
        if(strcmp(data1,"1") == 0){
            aktuator[pompa1]=1;
            digitalWrite(aktuatorPin[pompa1],LOW);
        }
        else{
            aktuator[pompa1]=0;
            digitalWrite(aktuatorPin[pompa1],HIGH);
        }
    }
}

```



```

}
Serial.println(data2);
if(mode[motor]==0){
    if(strcmp(data2,"1") == 0){
        aktuator[motor]=1;
        digitalWrite(aktuatorPin[motor],LOW);
    }
    else{
        aktuator[motor]=0;
        digitalWrite(aktuatorPin[motor],HIGH);
    }
}
Serial.println(data3);
if(mode[valve1]==0){
    if(strcmp(data3,"1") == 0){
        aktuator[valve1]=1;
        digitalWrite(aktuatorPin[valve1],LOW);
    }
    else{
        aktuator[valve1]=0;
        digitalWrite(aktuatorPin[valve1],HIGH);
    }
}
Serial.println(data4);
if(mode[valve2]==0){
    if(strcmp(data4,"1") == 0){
        aktuator[valve2]=1;
        digitalWrite(aktuatorPin[valve2],LOW);
    }
    else{
        aktuator[valve2]=0;
        digitalWrite(aktuatorPin[valve2],HIGH);
    }
}
Serial.println(data5);
if(mode[valve3]==0){
    if(strcmp(data5,"1") == 0){
        aktuator[valve3]=1;
        digitalWrite(aktuatorPin[valve3],LOW);
    }
}

```

```

else{
    aktuator[valve3]=0;
    digitalWrite(aktuatorPin[valve3],HIGH);
    }
    }
    buffer=0;
    }
}
client.stop();
}
aksesWebTime=millis();
}*/
delay(1000);
}

```

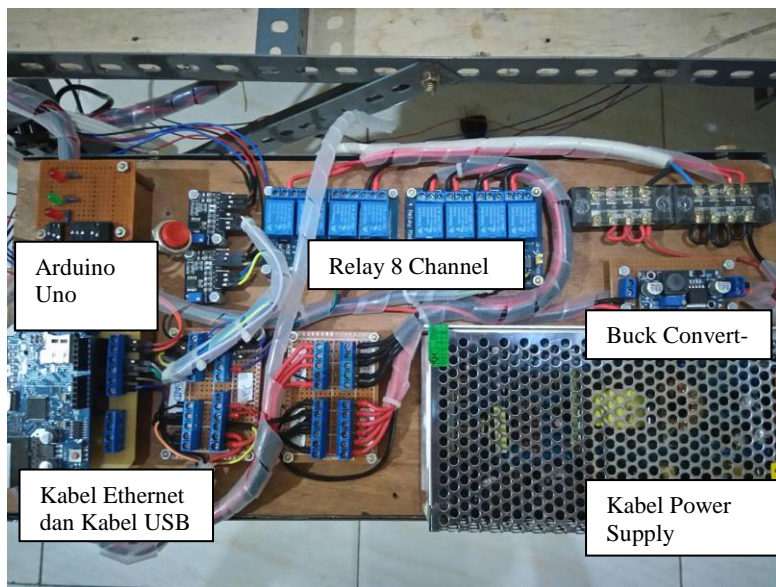
A.2 Lampiran Program Pengujian Pompa Air dan Relay

```
int ketinggian;  
int Pompa1;  
void setup(){  
  Serial.begin(9600);  
  pinMode(Pompa1, OUTPUT);}
  
void loop(){  
  if (ketinggian<25) {  
    (digitalWrite(Pompa1, HIGH));  
  }  
  if (ketinggian =25){  
    (digitalWrite(Pompa1, LOW));  
  }  
  else {  
    digitalWrite(Pompa1,LOW);}}
```

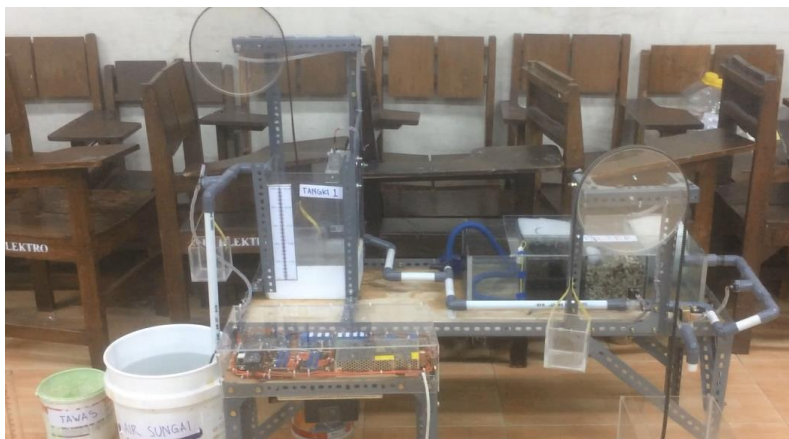
-----Halaman ini sengaja dikosongkan-----

LAMPIRAN B

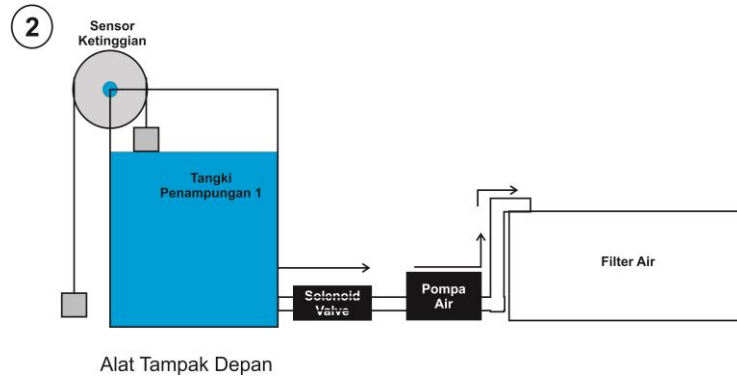
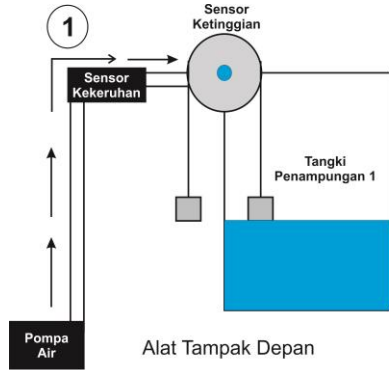
B.1 Kontroler Alat



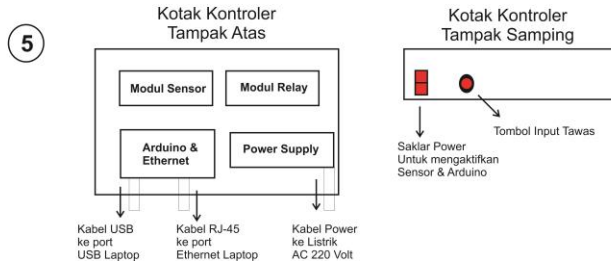
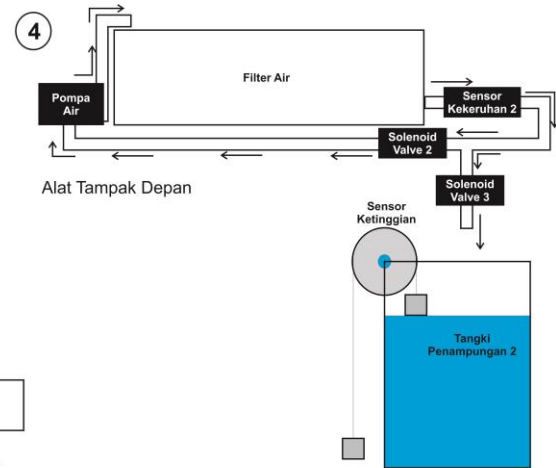
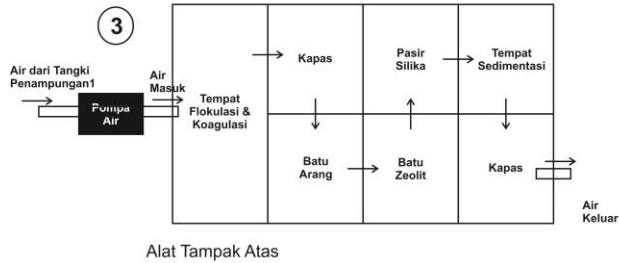
B.2 Alat Telemetry Penjernihan



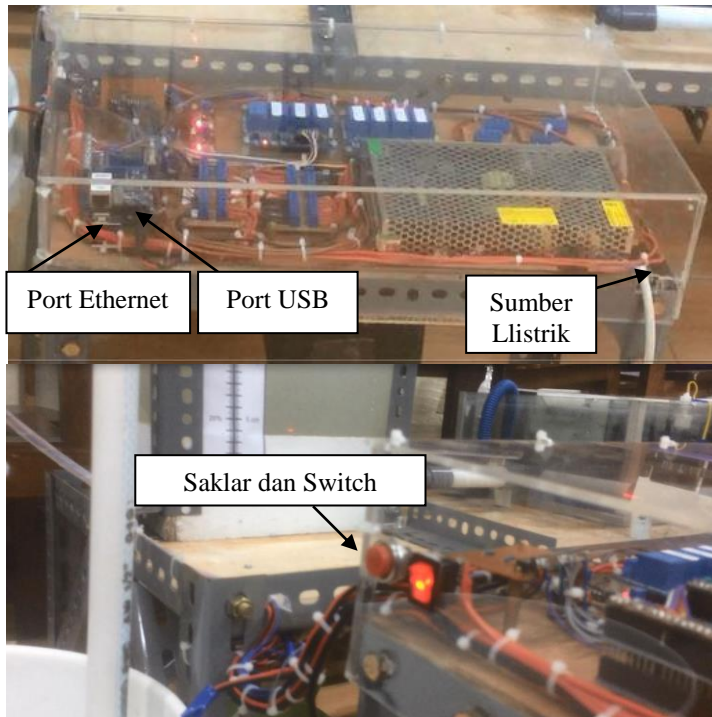
B.3 Alur Kerja dari Alat Sebelum Filter



B.4 Alur Kerja dari Alat Setelah Filter



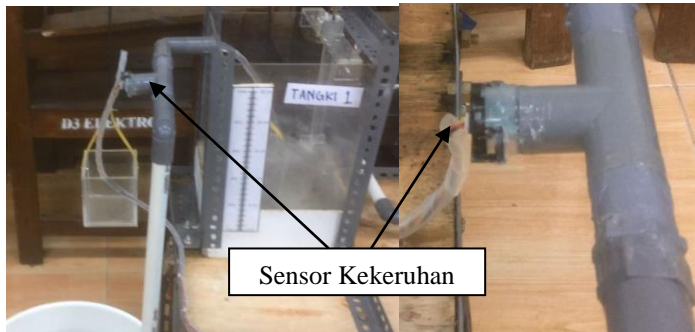
B.5 Kotak Kontroler



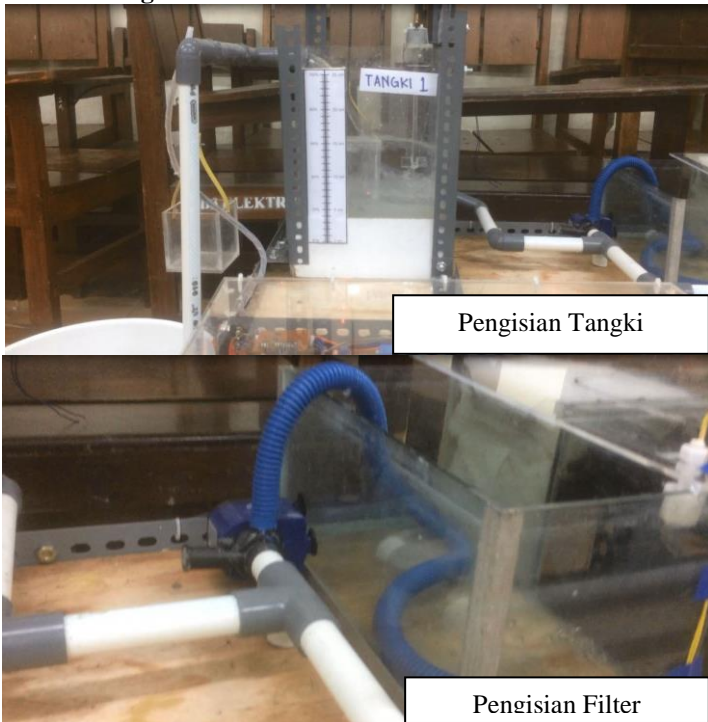
B.6 Filter Aquarium



B.7 Sensor Kekeruhan Pada Alat



B.8 Proses Pengisian



-----Halaman ini sengaja dikosongkan-----

LAMPIRAN C

DAFTAR RIWAYAT HIDUP



Nama : Rivaldy Hariansyah
TTL : Surabaya, 15 Mei 1996
Jenis Kelamin : Laki - Laki
Agama : Islam
Alamat : Jl.Simo Tambahan
Sekolahan 143, Surabaya
Telp/HP : 082234437649
E-mail : rivaldyhaiansyah14@gmail.com

RIWAYAT PENDIDIKAN

- 1. 2001 - 2008 : SD Negeri 02 Simomulyo Surabaya***
- 2. 2008 - 2011 : SMP Negeri 03 Surabaya***
- 3. 2011 - 2014 : SMA Negeri 06 Surabaya***
- 4. 2014 - 2017 : D3 Teknik Elektro, Fakultas Vokasi - FV
Insti-tut Teknologi Sepuluh Nopember (ITS)***

PENGALAMAN KERJA

- Kerja Praktek di PT Petrokimia Gresik

PENGALAMAN ORGANISASI

- Asisten Lab Automation and Computer Laboratory 2015/2016

-----Halaman ini sengaja dikosongkan-----

DAFTAR RIWAYAT HIDUP



Nama : Achmad Luki Ssatriawan
TTL : Gresik, 12 November 1995
Jenis Kelamin : Laki-laki
Agama : Islam
Alamat : Jl. Gebang Kidul 22B
Telp/HP : 087753379505
E-mail : lukisatriawan@gmail.com

RIWAYAT PENDIDIKAN

1. **2001 – 2008 : SDN Pongangan II Manyar Gresik**
2. **2008 – 2011 : SMPN 1 Manyar Gresik**
3. **2011 – 2014 : SMA Muhammadiyah 1 Gresik**
4. **2014 – 2017 : D3 Teknik Elektro, Fakultas Vokasi - FV Institut Teknologi Sepuluh Nopember (ITS)**

PENGALAMAN KERJA

1. Kerja Praktek di PT Petrokimia Gresik

PENGALAMAN ORGANISASI

1. Asisten Lab Automation and Computer Laboratory 2015/2016
2. Staff Dagri HIMAD3TEKTRO 2015/2016

-----Halaman ini sengaja dikosongkan-----